



StrategyQuant

User's Guide

Support

If you'll have trouble understanding anything, you need help, or you simply have some question to ask (related to the system), remember your purchase includes also a support.

We are here for you, you can contact us at:

www.StrategyQuant.com/contactus

Articles and forum on our website provide additional source of knowledge

www.StrategyQuant.com/articles

www.StrategyQuant.com/forum

Copyright

All rights reserved.

StrategyQuant software, bonus strategies and content of this Manual is copyrighted.
You can use them only with valid license.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means - including electronic, mechanical, photocopy, recording, scanning or otherwise - without the prior written permission of the author.

© 2012-2013 Mark Fric, SonarBytes Ltd

Risk Disclosure

Risk Disclosure Statement

Trading any financial market involves risk. This Manual is neither a solicitation nor an offer to Buy/Sell any financial product. The contents of this Manual are for general informational purposes only.

Although every attempt has been made to ensure accuracy, the author does not give any expressed or implied warranty as to its accuracy. The author does not accept any liability for error or omission. All examples are provided for illustrative purposes only and should not be construed as investment advice.

No representation is being made that any account, or trader will, or is likely to achieve profits or losses similar to those discussed in this Manual. Past performance cannot be relied upon as being indicative of future performance.

The information provided in this Manual is not intended for distribution to, or use by any person or entity in any jurisdiction or country where such distribution or use would be contrary to law or regulation or which would subject the author to any registration requirement within such jurisdiction or country.

Hypothetical performance results have many inherent limitations, some of which are mentioned below. No representation is being made that any account will or is likely to achieve profits or losses similar to those shown. In fact, there are frequently sharp differences between hypothetical performance results and actual results subsequently achieved by any particular trading system.

One of the limitations of hypothetical performance results is that they are generally prepared with the benefit of hindsight. In addition, hypothetical trading does not involve financial risk and no hypothetical trading record can completely account for the impact of financial risk in actual trading.

For example the ability to withstand losses or to adhere to a particular trading program in spite of the trading losses are material points, which can also adversely affect trading results. There are numerous other factors related to the market in general or to the implementation of any specific trading program, which cannot be fully accounted for in the preparation of hypothetical performance results. All of which can adversely affect actual trading results.

U.S. Government Required Disclaimer - Commodity Futures Trading Commission Futures, Currency and Options trading has large potential rewards, but also large potential risk. You must be aware of the risks and be willing to accept them in order to invest in the futures and options markets. Don't trade with money you can't afford to lose. This is neither a solicitation nor an offer to Buy/Sell futures or options. No representation is being made that any account will or is likely to achieve profits or losses similar to those discussed on this web site. The past performance of any trading system or methodology is not necessarily indicative of future results.

CFTC RULE 4.41 - HYPOTHETICAL OR SIMULATED PERFORMANCE RESULTS HAVE CERTAIN LIMITATIONS. UNLIKE AN ACTUAL PERFORMANCE RECORD, SIMULATED RESULTS DO NOT REPRESENT ACTUAL TRADING. ALSO, SINCE THE TRADES HAVE NOT BEEN EXECUTED, THE RESULTS MAY HAVE UNDER-OR-OVER COMPENSATED FOR THE IMPACT, IF ANY, OF CERTAIN MARKET FACTORS, SUCH AS LACK OF LIQUIDITY. SIMULATED TRADING PROGRAMS IN GENERAL ARE ALSO SUBJECT TO THE FACT THAT THEY ARE DESIGNED WITH THE BENEFIT OF HINDSIGHT. NO REPRESENTATION IS BEING MADE THAT ANY ACCOUNT WILL OR IS LIKELY TO ACHIEVE PROFIT OR LOSSES SIMILAR TO THOSE SHOWN.

Software License Agreement

This legal document is an agreement between you, the end user ('User'), and Sonarbytes Ltd ('Author').

AGREEMENT - By installing StrategyQuant ('Software'), copying the Software and/or clicking on the 'I Agree' button during installation, you agree to all of the terms of this software license agreement ('Agreement'). If you do not agree with all of the terms of this agreement, click on the 'I Do Not Agree' button and/or do not install, copy or otherwise use the software.

INTELLECTUAL PROPERTY. The Software and any associated materials are protected by copyright law. The package is a proprietary product of the Author. The Author retains title to and ownership in the copyright of the software program and the associated materials. You acknowledge that the Author owns all rights, title and interest in and to the Software, including without limitation all Intellectual Property Rights. 'Intellectual Property Rights' means any and all rights existing from time to time under patent law, copyright law, trade secret law, trademark law, unfair competition law, and any and all other proprietary rights, and any and all applications, renewals, extensions and restorations thereof, now or hereafter in force and effect worldwide. You agree not to modify, adapt, translate, decompile, reverse engineer, disassemble or otherwise attempt to derive source code from the Software.

REGISTRATION. This program is neither freeware nor public domain. Use requires valid license. Contact us at <http://www.geneticbuilder.com/contactus/> for bulk licenses or discounted prices.

GRANT. Author hereby grants you a non-exclusive, non-transferable license to use the Software upon payment of the License Fee until the expiry date of the license (if any). Author makes no guarantee of the frequency, value, applicability or content of future updates or modifications to the Software. The Software will only be made available to you in electronic form for download.

The requirement to pay a license fee does not apply to evaluation copies for which Author does not charge a license fee. Evaluation licenses expire 14 calendar days from the date of this agreement, unless otherwise agreed to in writing by Author. On the date of expiry of the license, User agrees to either purchase the Software at the list price in force at that time or to destroy all copies of the Software in electronic or other form, including any copies on backup tapes or other media.

User's use of the Software shall be limited to use on a single hardware chassis, on a single central processing unit, as applicable, or use on such greater number of chassis or central processing units as User may have paid the required license fee.

You may not: permit other individuals to use the Software except under the terms listed above; translate, reverse engineer, decompile, decrypt, reverse engineer, disassemble (except to the extent applicable laws specifically prohibit such restriction), or create derivative works based on the Software; copy the Software (except for back-up purposes); rent, lease, transfer, assign, sub-license or otherwise transfer rights to the Software; or remove any proprietary notices or labels on the Software.

DISCLAIMER OF WARRANTY. THE SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE IS BORNE BY YOU.

LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, TORT, CONTRACT, OR OTHERWISE, SHALL AUTHOR OR ITS SUPPLIERS OR RESELLERS BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL OR PUNITIVE DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES. IN NO EVENT WILL AUTHOR BE LIABLE FOR ANY DAMAGES IN EXCESS OF AUTHOR'S LIST PRICE FOR A LICENSE TO THE SOFTWARE, EVEN IF AUTHOR SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY

ANY OTHER PARTY. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. FURTHERMORE, SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION AND EXCLUSION MAY NOT APPLY TO YOU.

TERMINATION. This license will terminate automatically if you fail to comply with the limitations described above. On termination, you must destroy all copies of the Software in electronic or other form, including any copies on backup tapes or other media. Upon termination of this License for any reason, you shall have no right to refund of the whole or part of any License Fee paid.

Table of Contents

1	Introduction.....	8
1.1	What is StrategyQuant?	8
1.2	System Requirements.....	10
1.3	Installation.....	10
2	How does it work?.....	11
2.1	Random generation of trading strategies	11
2.2	Genetic Evolution	12
2.3	Supported building blocks	13
2.4	Custom Indicators.....	14
3	Quick start with the program	15
3.1	Main concepts	15
3.1.1	Layout	15
3.1.2	Databanks	16
3.1.3	Working with files.....	17
3.2	Flow of work	18
4	Program modes	20
4.1	Data Manager.....	20
4.1.1	History Data	20
4.1.2	Custom indicators.....	21
4.2	Build strategies	22
4.3	Retest strategies.....	23
4.4	Improve strategies.....	23
4.5	Optimizer	23
5	Program Layout	24
5.1	Progress	25
5.2	Settings.....	26
5.2.1	Data	26
5.2.2	Building blocks.....	29
5.2.3	Strategy options	30
5.2.4	Genetic options	32
5.2.5	Money management	34
5.2.6	Robustness tests.....	35
5.2.7	Ranking options	37
5.2.8	Parts to improve	39
5.2.9	Parameters	40
5.2.10	Success criteria	41
5.2.11	Global options	43
5.3	Results	44
5.3.1	Databanks.....	44
5.3.2	Overview.....	45
5.3.3	List of trades	45
5.3.4	Equity chart	45
5.3.5	Trade analysis	45
5.3.6	Robustness analysis.....	45
5.3.7	Strategy settings.....	46

5.3.8	Source code	47
6	Advanced functionality.....	48
6.1	Robustness tests and analysis	48
6.2	Optimization.....	53
6.2.1	Simple optimization.....	53
6.2.2	Walk-Forward Optimization	59
6.2.3	Walk-Forward Matrix	64
6.3	Portfolios	68
6.4	Strategy Editor.....	69
7	How to...	71
7.1	Import history data from MetaTrader	71
7.2	Import new custom indicator.....	74
7.3	Export strategy from StrategyQuant and test or trade it in MetaTrader.....	81
7.4	Translate program to another language	85
8	Appendix.....	86
8.1	Ranking criteria.....	86
8.2	Entry and exit types.....	90
9	Final Words.....	92

1 Introduction

1.1 What is StrategyQuant?

StrategyQuant is a program that automatically generates new unique trading strategies for forex, stocks or ETFs.

Using StrategyQuant, you can find profitable trading strategies for virtually any market, any timeframe and any chart type. No programming or trading knowledge is required.

The resulting strategies can be saved as a MetaTrader 4 Expert Advisor with complete source code.

With StrategyQuant you can:

- Build an unlimited number of **unique trading strategies**
- Develop strategies for virtually any market or timeframe
- Save your strategies as a **MetaTrader Expert Advisor with full source code!**
- Eliminate the manual labor previously required when developing a trading strategy
- **Find new trading strategies** that are not only unique, but also non-obvious
- **Reduce the time required to build a strategy** from weeks and months to minutes!
- Improve your existing strategies

Is StrategyQuant Right for You?

If you trade using automatic trading system (called also robots or Expert Advisors) or you plan to develop your own trading strategies then **StrategyQuant can save you money and hundreds of hours of your time.**

Some traders prefer to purchase an existing trading robot; there is a multitude of offers especially for Forex.

While this can be an effective way, by purchasing someone's forex robot you are usually purchasing a black box - you don't know how it works, what are the exact rules and you are at mercy of its creator with potential adaptations to changed market conditions.

StrategyQuant allows you to **create your own trading strategies**, exportable to a plain Expert Advisor source code, so you have full control over your strategy.

What If You Are a Manual Trader?

You can still use for StrategyQuant to generate the trading ideas. You'll be surprised to find many profitable strategies based on relatively simple rules that you wouldn't think of.

Every strategy that is created by StrategyQuant can be exported to readable pseudo code with full description of the trading rules and can be traded also manually.

What to Expect

Please keep in mind that **StrategyQuant is a powerful tool, but it is not a magic box** that will start making you money with a click on a button. It has to be used in the right way to get the results.

Generating new strategies in StrategyQuant is only about 50% of the work.

The rest of the work has to go into evaluating the generated strategies to filter out the ones that are curve fitted or not robust enough.

It is up to you to evaluate your new strategies properly and know their strengths, weaknesses and limitations before you put them to live trading.

It can easily happen that from all the profitable strategies generated by StrategyQuant **only 1 out of 10 passes the evaluation** and we can consider using it for live trading.

But - the number of strategies we can generate is almost endless, so even 5-10% from infinity is a pretty big number :-)

There are few steps to evaluate the strategies quality and measure how good they will be in real live trading. Please read the **Evaluating Generated Strategies** section for more information.

1.2 System Requirements

StrategyQuant is not an EA, it is a normal program (EXE file) for Microsoft Windows and it will run on all standard computers **with Internet connection**.

Minimum system requirements:

- 1.2 GHz processor
- 512 MB RAM
- 500 MB hard disk space
- Windows XP, Vista, Windows 7 or Windows 8 operating system

Computing power

Strategy generation and backtesting requires a lot of processing power, so the faster computer you have, the more strategies will be generated and tested.

The most important component that affects the speed of testing is processor (CPU). StrategyQuant is able to use all processor cores, so the best results will be achieved by using multi-core processor, such as i5 or i7.

1.3 Installation

StrategyQuant comes with standard setup wizard, you just download and run the installation EXE file and then follow the steps in the installation wizard.

Important !!!

Please DON'T install StrategyQuant to standard C:\Program Files directory !

It might not work correctly because Windows security settings don't allow the program to write to its data files.

Instead install it to any normal drive or directory on the disk,
like C:\StrategyQuant
or C:\Trading\StrategyQuant

2 How does it work?

StrategyQuant is a program, it doesn't have the brain or experience of a trader, and it doesn't know how to create a profitable strategy.

What it does is that it randomly combines available building blocks (indicators, prices, etc.) to create new trading rules. The resulting strategy is then tested on a history data to see if it is profitable.

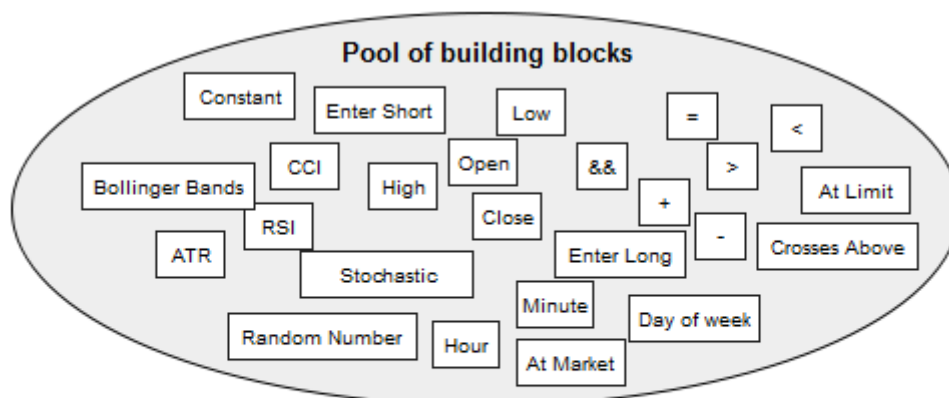
Random generation is the foundation of StrategyQuant. Strategies generated this way can be further improved (evolved) using **Genetic evolution**.

2.1 Random generation of trading strategies

A trading strategy in the initial population is constructed using a combination of price patterns, technical indicators, order types, and other parts to form the entry and exit rules.

StrategyQuant can use all standard technical indicators and oscillators (like CCI, RSI, Stochastic, etc.), time values (like time of day, day of week) and price patterns. These building blocks are then combined using logical and equality operators (and, or, >, <, etc.) to form an entry or exit rule. In addition, it supports different entry and exit order types (market order, limit order, fixed profit target, exit after X bars, etc.).

With all the possible combinations of rules and orders, StrategyQuant is capable of generating literally trillions of different possible trading strategies.



Example of randomly generated entry rule:

CCI	60	>	0	&&	Day of week	<>	Monday	Enter Long	At Market
-----	----	---	---	----	-------------	----	--------	------------	-----------

if (CCI(60) > 0 && Day of week <> Monday) then Enter Long At Market

The building process itself is completely random - builder randomly picks different building blocks from the available pool and combines them to create entry rule, order type and exit rule.

There are some validity constraints that ensure that, for example price is not compared to time value, etc.

The result is a completely new random trading strategy. Of course, not every randomly created

strategy is profitable, but StrategyQuant can produce and test thousands of new strategies per hour, and there are many profitable ones in this amount.

2.2 Genetic Evolution

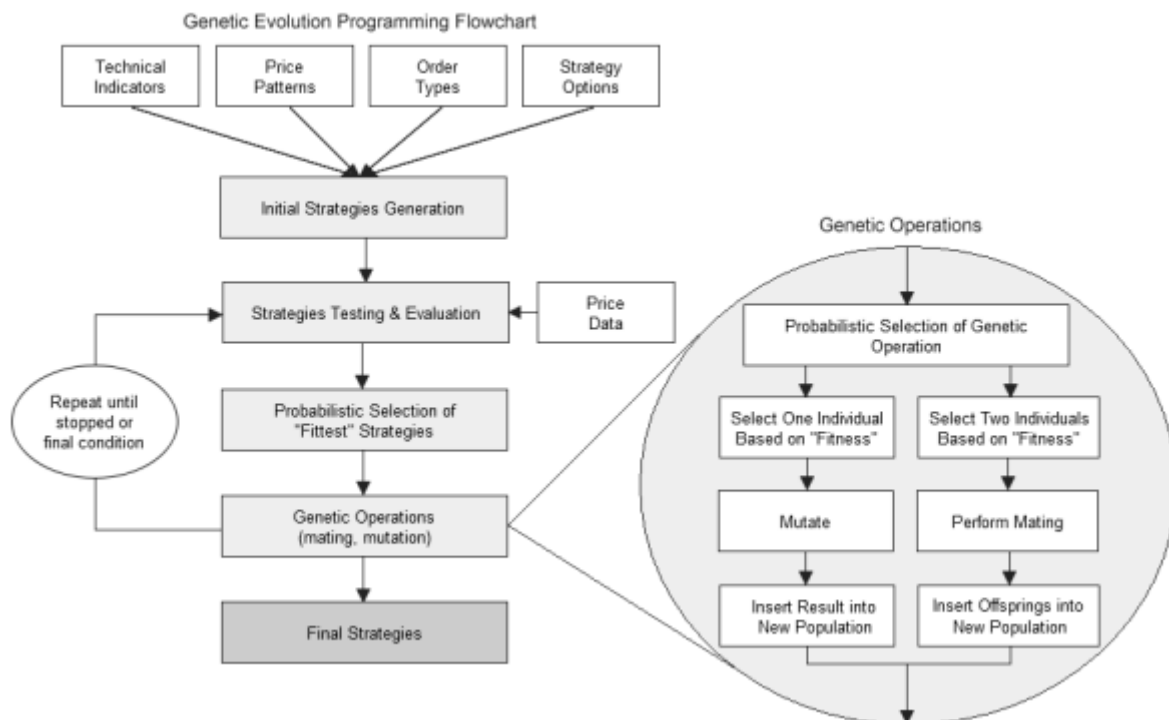
Genetic Evolution takes the process of finding suitable trading strategies even further.

In this mode StrategyQuant first creates a number of random strategies, which are used as the initial population in the evolution.

This initial generation of strategies is then "evolved" over successive generations using genetic programming technology.

This process imitates the evolution - the algorithm chooses the fittest strategies (using selected performance criteria) in every generation, and the group of fittest candidates is then used to produce new generation of trading strategies.

As in evolution, this should result in better and better candidates, in our case in strategies that are more profitable, more stable, or generally better in the selected performance criteria.



2.3 Supported building blocks

StrategyQuant currently supports the following components to build entry and exit rules:

Indicators

Simple Moving Average	Average Directional Movement Index (ADX)
Exponential Moving Average	Average True Range (ATR)
Weighted Moving Average	Momentum
Commodity Channel Index (CCI)	Williams % Range
Relative Strength Index (RSI)	True Range
Stochastic	Price Difference
MACD	Highest, Lowest
Bollinger Bands	Keltner Channel
Qualitative Quantitative Estimation (QQE)	Parabolic SAR
Triple Exponential Moving Average	Ichimoku
Custom Indicators ***	

Price Values

Open	Low
High	Close
Open Daily	Low Daily
High Daily	Close Daily
Heiken Ashi Open	Heiken Ashi Low
Heiken Ashi High	Heiken Ashi Close

Candle Patterns

Doji	Shooting Star
Hammer	Dark Cloud Cover
Bullish Engulfing	Piercing Line
	Bearish Engulfing

Operators

Greater	Addition (+)
Lower	Subtraction (-)
Crosses Above	Multiplication (*)
Crosses Below	Equals
And	Not Equals
Or	

Time Values

Hour	Day of Week
Minute	

Order Types

Enter at Market	Enter at Stop
Enter/Reverse at Market	Enter at Limit

Exit Types

Stop Loss	Profit Target
Exit After X Bars	Move Stop Loss to Break-even
Exit Rule (Price + Operators + Indicators, ...)	Stop Trailing
	Profit Trailing

We will be continually adding new technical indicators and other features to the Builder. If you have your favorite indicator you'd like to see in StrategyQuant, just let us know.

2.4 Custom Indicators

Since version 2.0 the program offers almost infinite flexibility of building blocks with custom indicators. In contrast with build-in indicators, StrategyQuant doesn't know how to compute custom indicators; they are defined by their data. This will enable you to use virtually any of your favorite indicators in StrategyQuant. There are no real limits – you can use multi-timeframe or multi-symbol indicators, generally every indicator that works in MetaTrader can be

The way it works is that you have to export the values of the indicator computed on a specific symbol and timeframe to a file, and then import this file as a custom indicator to the Builder. This way you'll create new custom indicator (with values computed in another program, such as MT4) for this specific symbol and timeframe also in SQ.

Remember, every custom indicator is defined for a specific combination of symbol and timeframe (on which it was computed).

To use the same indicator on another symbol or timeframe, you have to recompute its data in MT4, export it and create a new custom indicator for this symbol/timeframe combination.

You can find more information about custom indicators in these topics:

[Manage Custom indicators](#)

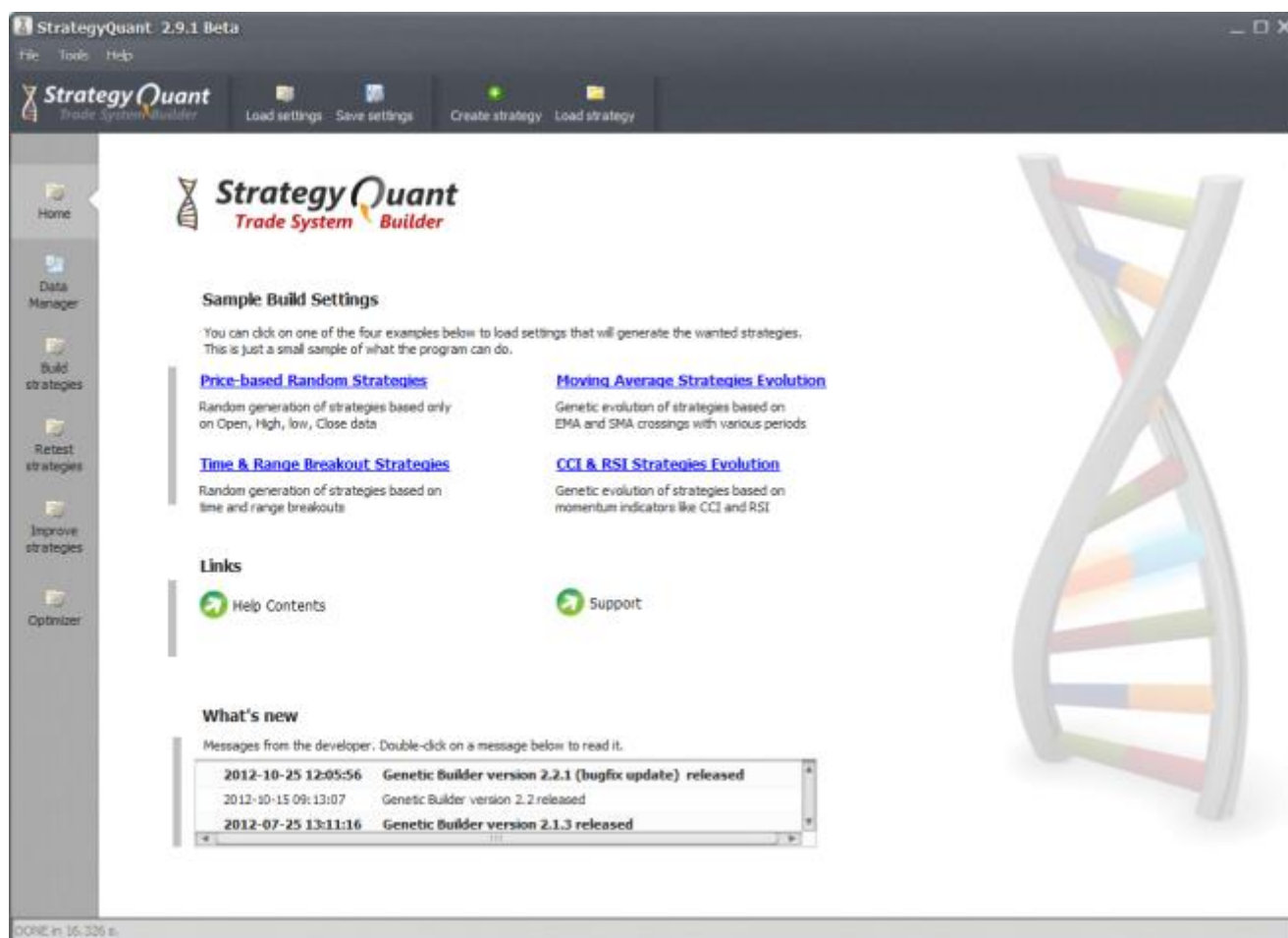
[How to ... Import new custom indicator](#)

3 Quick start with the program

3.1 Main concepts

3.1.1 Layout

When you'll first start the program you'll see the main screen as in the picture below.



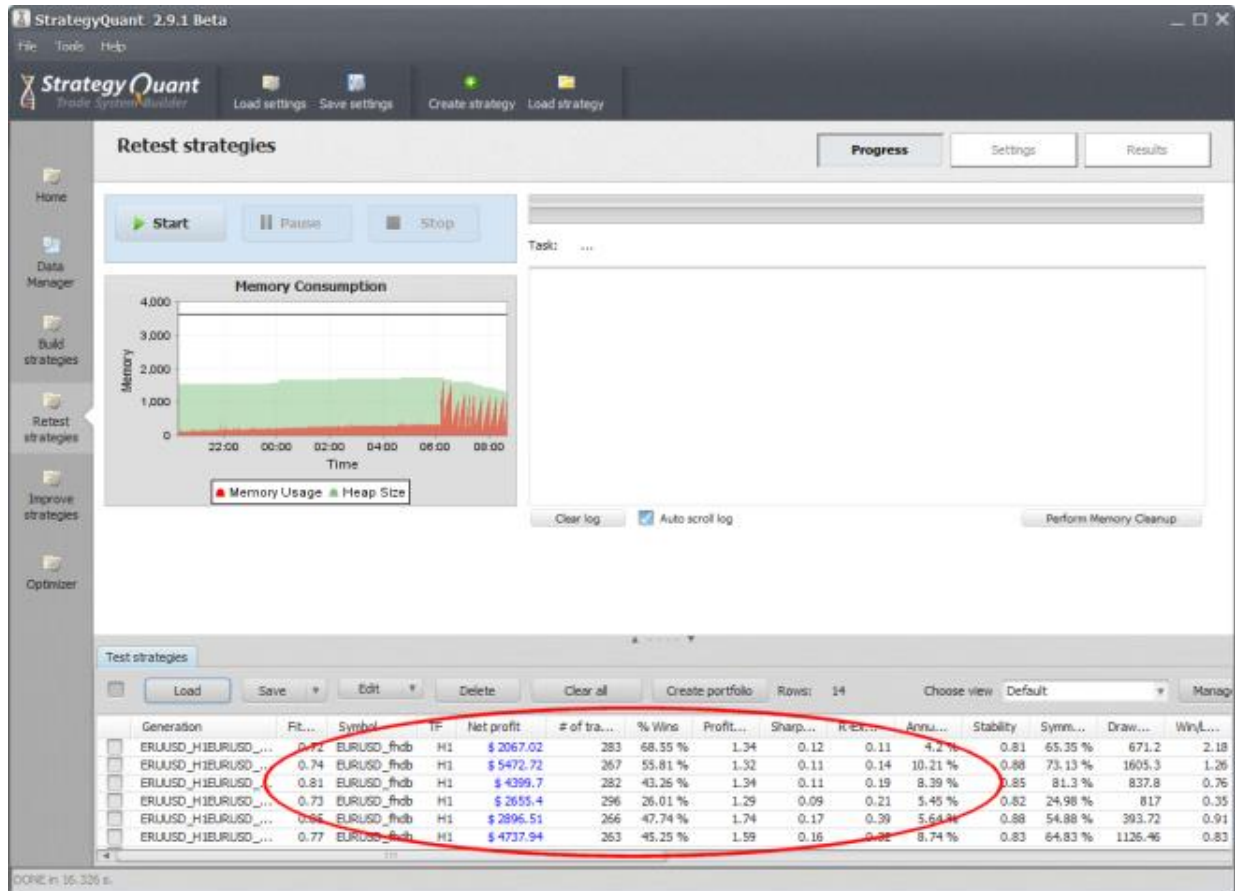
The program functionality is divided into tabs on the left side.

These tabs are:

- **Home** - starting screen of the program, it contains the sample settings, news and helpful links
- [Data manager](#)
- [Build strategies](#)
- [Retest strategies](#)
- [Improve strategies](#)
- [Optimizer](#)

3.1.2 Databanks

Databank is the most important concept you should understand when using StrategyQuant. Doesn't matter which mode you use, the best resulting strategies are always stored in the Databank.



There you can sort the strategies by its properties, load and save them, and when you double click on a row, it will open the strategy test details in the results window above.

You can configure how many top strategies (10, 100 or 500) will be stored in Databank, what comparison criterion to use which strategies to dismiss automatically (for example those with negative P/L).

3.1.3 Working with files

Strategies in Genetic Builder are saved in its own proprietary file format (with .str extension) that can be opened only by StrategyQuant.

If you find potentially good strategies you should always save them so that you can work with them later.

Running strategies in MetaTrader

MetaTrader cannot read the strategy .str files. If you want to test or run your new strategies in MetaTrader you have to export the strategy to MQL source code.

Please check the section [How To ... Export strategy from StrategyQuant and test it in MetaTrader](#).

Please note that exported (.MQ4) files are not readable by StrategyQuant, so make sure you always save your strategy also as a normal strategy file (*.str).*

3.2 Flow of work

The general flow of work when generating new strategies can be described as follows:

1. Configure data for backtest

You can use the history data that come with the program or optionally import your own data in MetaTrader format. Then set up the time range and timeframe you want to use. If you'll use genetic evolution mode you should divide the data to In Sample and Out of Sample periods.

You can also use additional data or Robustness tests to automatically test the strategy robustness.

2. Configure settings

Go through all the settings and configure strategy type, indicators and order types to be used for trading rules. Optionally use time constraints to limit trading to a certain time range.

3. Configure Ranking options

Ranking options allow you to select Strategy Selection Criteria - which is how the best strategies are determined.

You should also set up Custom Conditions to filter only strategies that pass certain criteria. It makes sense to dismiss all the strategies that have too little profit or trades, or too small Profit Factor, Return/DD ratio or System Quality Number.

4. Run Build

Start the building process. Depending on your settings you can let it run several minutes, several hours or even several days. The more time it will run the more potential strategies it will test. The best of them will be stored into the databank.

5. Evaluate generated strategies

Go through the generated strategies and evaluate them. You can either evaluate them visually by checking their equity chart, or by sorting them by their parameters in the Databank. Choose the best ones to pass to the next step and save them to StrategyQuant project file (.sqn) so that you can work with them later.

Evaluation can consist also of [Retesting](#) the strategies on additional symbols and/or timeframes, or with different spread and slippage and comparing the results.

You should also run [Robustness Tests](#).

The goal of strategy evaluation is to find strategies that are robust - that means they work in different conditions

and don't break up when there is a small change in parameters or in price data or they miss few trades.

Optional additional steps are:

6. Improve the strategy

You can try to improve the strategy in Improver. You can try to apply different combinations of exit rules or additional conditions to entry rules in search for better performance.

After the improvement you should again run the new strategy variation through robustness tests to make sure it didn't lose its robustness.

6. Optimize the strategy

You can run [simple optimization](#) to find better combination of input parameters of your strategy. You can also run [Walk-Forward optimization](#) to find out if the strategy would benefit from periodical reoptimization.

As the last step, you can run [Walk-Forward Matrix](#) analysis to determine the best reoptimization period.

There are articles on our website that describe more detailed flow of work:

<http://www.strategyquant.com/articles/getting%20started%20with%20StrategyQuant>

<http://www.strategyquant.com/articles/strategy%20building%20process>

4 Program modes

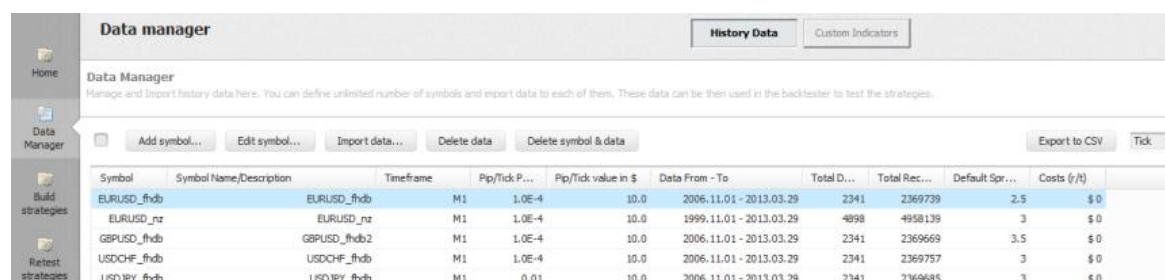
The program is divided into four independent (+ data manager) sections:

4.1 Data Manager

a place where you can manage history data and custom indicators used in the program.

4.1.1 History Data

This screen is where you can manage (create, import, modify) the history data that are used for backtesting.



The screenshot shows the 'Data manager' window with the 'History Data' tab selected. Below the tab are buttons for 'Add symbol...', 'Edit symbol...', 'Import data...', 'Delete data', and 'Delete symbol & data'. On the right, there are 'Export to CSV' and 'Tick' buttons. The main table lists symbols, their descriptions, timeframes, pip/tick values, and data ranges.

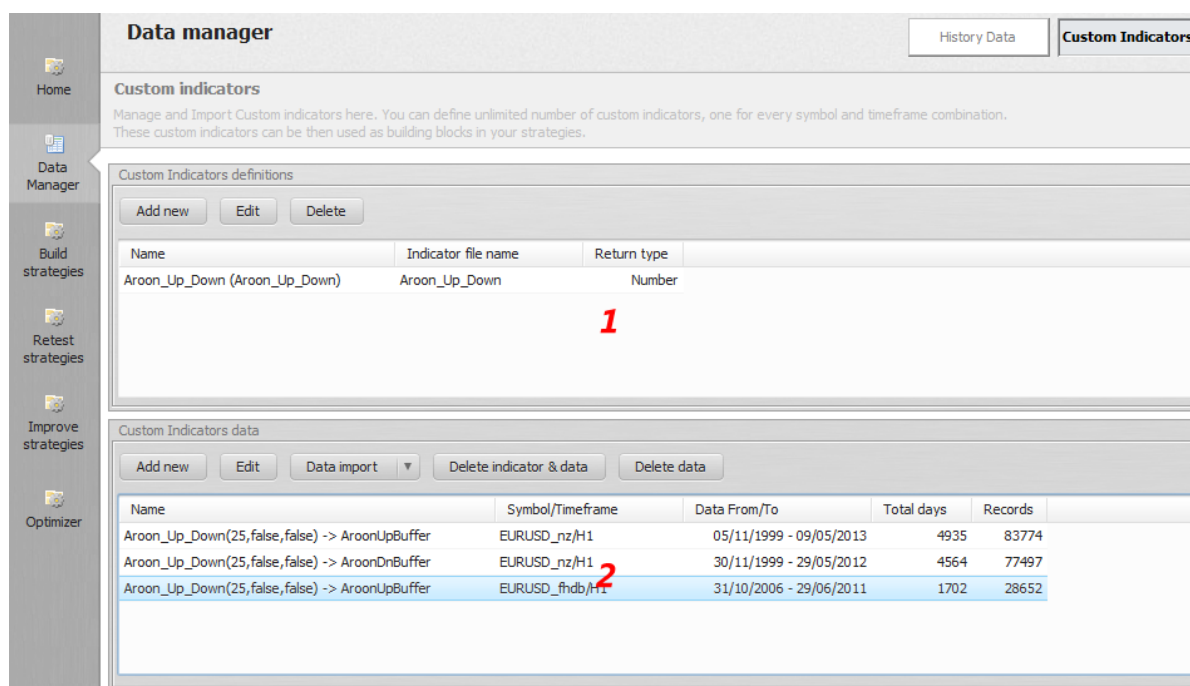
Symbol	Symbol Name/Description	Timeframe	Pip/Tick P...	Pip/Tick value in \$	Data From - To	Total O...	Total Rec...	Default Spr...	Costs (r/t)
EURUSD_fhdb	EURUSD_fhdb	M1	1.0E-4	10.0	2006.11.01 - 2013.03.29	2341	2369739	2.5	\$ 0
EURUSD_nr	EURUSD_nr	M1	1.0E-4	10.0	1999.11.01 - 2013.03.29	4898	4958139	3	\$ 0
GBPUSD_fhdb	GBPUSD_fhdb2	M1	1.0E-4	10.0	2006.11.01 - 2013.03.29	2341	2369669	3.5	\$ 0
USDCHF_fhdb	USDCHF_fhdb	M1	1.0E-4	10.0	2006.11.01 - 2013.03.29	2341	2369757	3	\$ 0
USDJPY_fhdb	USDJPY_fhdb	M1	0.01	10.0	2006.11.01 - 2013.03.29	2341	2369685	3	\$ 0

StrategyQuant already comes with 4 years history data for the major pairs provided by ForexHistoryDatabase.com, but if you have your own history data in Metatrader, you can easily import your data

Check the section [Import history data from MetaTrader](#) for more description.

4.1.2 Custom indicators

In this screen you can manage and import custom indicators to be used in StrategyQuant.



Data manager History Data Custom Indicators

Custom indicators
Manage and Import Custom indicators here. You can define unlimited number of custom indicators, one for every symbol and timeframe combination. These custom indicators can be then used as building blocks in your strategies.

Custom Indicators definitions

Add new Edit Delete

Name	Indicator file name	Return type
Aroon_Up_Down (Aroon_Up_Down)	Aroon_Up_Down	Number

1

Custom Indicators data

Add new Edit Data import Delete indicator & data Delete data

Name	Symbol/Timeframe	Data From/To	Total days	Records
Aroon_Up_Down(25,false,false) -> AroonUpBuffer	EURUSD_nz/H1	05/11/1999 - 09/05/2013	4935	83774
Aroon_Up_Down(25,false,false) -> AroonDnBuffer	EURUSD_nz/H1	30/11/1999 - 29/05/2012	4564	77497
Aroon_Up_Down(25,false,false) -> AroonUpBuffer	EURUSD_fhdb/H1	31/10/2006 - 29/06/2011	1702	28652

2

The screen is divided into two tables. Upper table (1) contains list of all custom indicators that are defined in the system.

Bottom table (2) contains indicator data - data for an indicator with given parameters, output values and symbol/timeframe.

It is important to understand that StrategyQuant is not able to run indicator code, so the only way how to work with custom indicators in SQ is by importing their data.

Data are dependent on indicator parameter values as well as on symbol and timeframe on which it was generated.

To find out more about how to work with custom indicators, check the section [How to ... Import new custom indicator](#)

4.2 Build strategies

the heart of the program. Here you can generate new trading strategies using different configuration options and building blocks.

The resulting strategies should be saved to StrategyQuant file (.str) so you can work with them later.

Available build modes:

Genetic Evolution

StrategyQuant first generates initial population of random candidates (using the Random Generation mode) and then uses genetic evolution process

to evolve the population and produce better and better candidates with each generation.

The process ends when predefined number of generations is reached or when there's no further improvement.

Pros:

- in theory it should lead to strategies better than the initial random generation
- this means that the already good strategies in the first generation can be further improved
- search for profitable strategy in the trillions of possible combinations can be more effective with the power of evolution

Cons:

- evolution can be slower
- sometimes the evolution can lead to the dead end, so the generation should be watched
- the group of generated strategies is limited by population size

Random Generation

In this mode StrategyQuant continually generates and tests new random strategies, one after another, until it is stopped.

The top candidates (based on predefined criteria) are stored into Databank so you can review them later.

Pros:

- faster and simpler than genetic evolution
- it will run until it is stopped, so if you let it run for a few days it can generate and evaluate millions of strategies

Cons:

- once the strategies are generated they are not further improved

4.3 Retest strategies

this section is used for retesting already existing strategies. You can retest the strategy on another historical data or timeframe; run Robustness tests on the strategy, try different options (for example trading only during certain hours per day).

4.4 Improve strategies

this section can be used to improve existing strategy created in the Build step. "Improving" strategy means replacing some of its part (entry rule, order type, exit rule) or adding new condition to make strategy "better".

Improver works in a way that it randomly generates the selected part of the strategy, while keeping the other strategy parts intact.

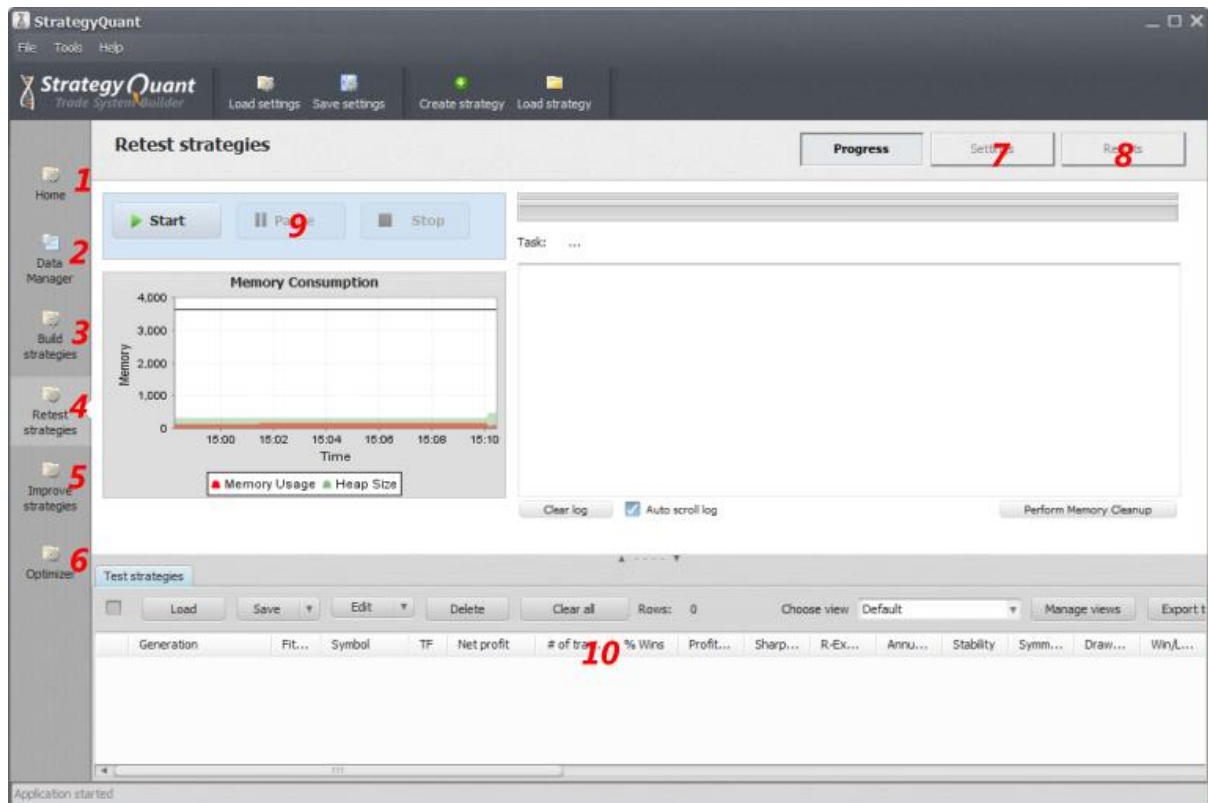
This way you can look for better exit rule or for additional entry rule condition that will improve strategy performance.

4.5 Optimizer

optimizer can be used to optimize strategy parameters to optimal values, as well as for Walk-Forward optimization and Walk-Forward Matrix analysis.

5 Program Layout

The program is divided into several independent sections that correspond with the main functionality.

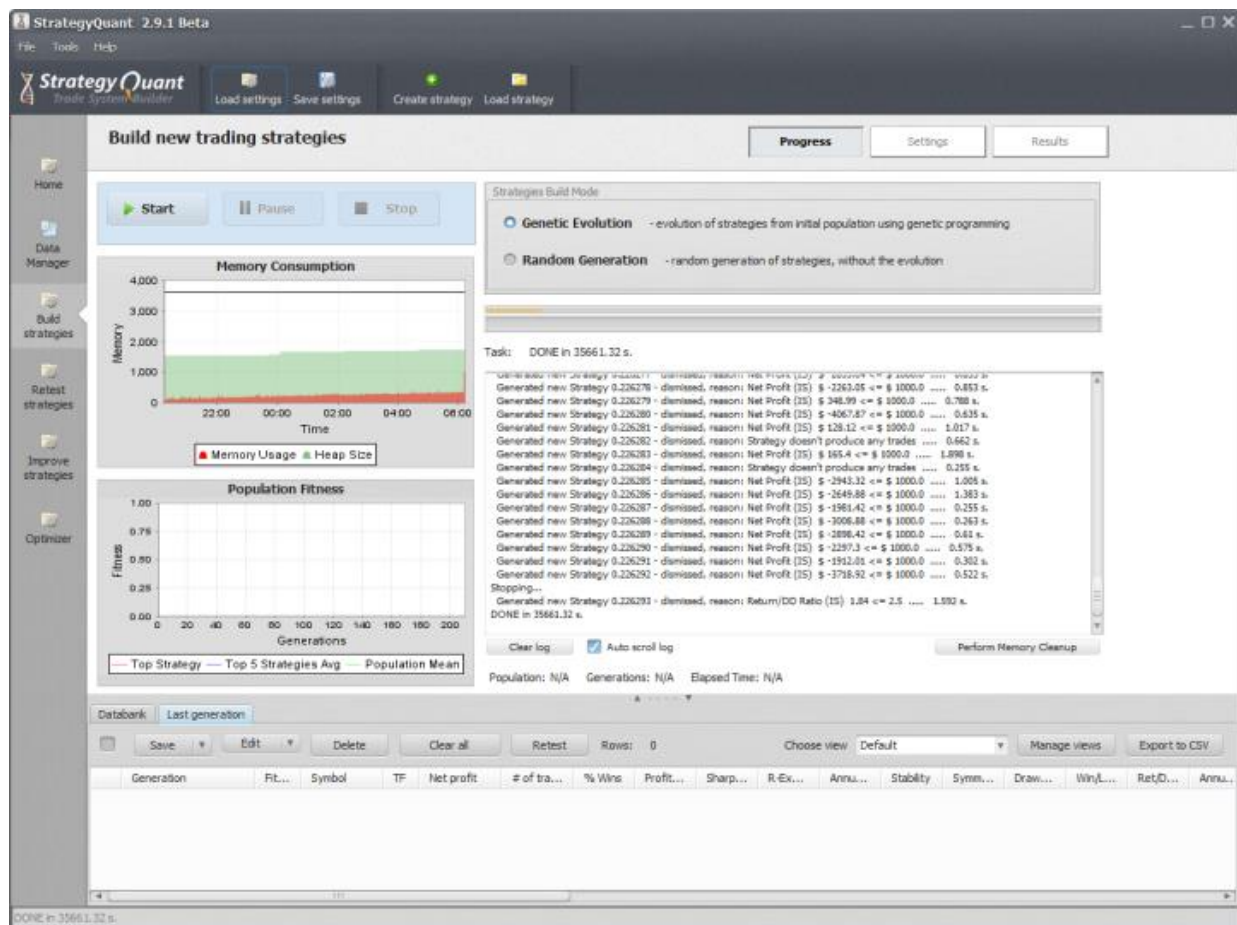


1. [Home](#)
starting screen of the program, it contains sample settings, news and helpful links
2. [Data Manager](#)
here you can manage your history data and custom indicators
3. [Build strategies](#)
section to build new trading strategies
4. [Retest strategies](#)
section to retest existing strategies with different settings, or on different symbols/timeframes
5. [Improve strategies](#)
section to improve existing strategy by generating additional rule conditions
6. [Optimizer](#)
with features of simple and Walk-Forward optimization and Walk-Forward Matrix
7. [Settings area](#)
each section has its own independent settings area where you can configure the build/test process
8. [Results area](#)
here you can see performance of the selected strategy and get its EA code

9. buttons for starting, pausing and stopping the process
10. databanks - lists where generated strategies are stored
storage area for strategies, it contains all the built / tested strategies

5.1 Progress

Progress screen is the place where you can start, pause or stop the building/testing process, it also shows you the log and memory consumption.



Before starting the build or test you should have the data and settings configured. The results (generated top strategies) will be continually stored in the **databank** on the bottom.

5.2 Settings

5.2.1 Data

in Data you have to select the symbol, timeframe and time range on which the strategies will be backtested.

Data for backtest

Choose program data source that will be used for backtests of the strategies. Data source allows you to configure also the In Sample and Out of Sample periods, for the selected symbol you can choose also default spread, slippage and precision used for testing.

Main data for testing

Symbol: GBPUSD_fhdb Period: H1 Available Data: 01/11/2006 - 30/06/2011

Start Date: 01/11/2006 End Date: 30/06/2011 Reset Dates

Out of Sample Period: In Sample: 01/11/2006 - 01/04/2010

☐ Reverse Out of Sample / In Sample Out Of Sample: 01/04/2010 - 30/06/2011

Test Parameters

Spread: 3.5 Market Slippage: 0 Min distance of order from price: 3

Test precision: Selected Timeframe only (fastest) ☐ Use real spread (only for real tick data)

Main data for testing

StrategyQuant tests all the generated strategies on a given history data and this screen allows you to choose the symbol and timeframe to be used for backtests.

There is an option to choose also the testing period (Date From, Date To).

Out of Sample Period allows you to split the history data to two parts:

- **In Sample** - it is used during genetic evolution to compute the rank of the strategies, so that the program is able to compare them
- **Out of Sample** - this part of the data is used to verify if strategy really works as expected also on data that it was not evolved

Note

Setting the Out of Sample period **makes sense for Genetic Evolution mode**, because only in this mode the fitness results computed in the In Sample data are used to rank the strategies to find the "fittest" strategies.

The other modes (Random Generation, Strategy Testing) don't use evolution, so you don't need to split the data to two periods.

For these two modes you can move the Out of Sample splitter to the far right, leaving Out of Sample period empty.

Test parameters

Testing precision

testing precision means how is the price data simulated during the backtest.

It is usually sufficient to use Selected Timeframe mode for Market orders, or Tick simulation mode for Stop/Limit orders.

You can also use faster mode for generation, and then slower mode for strategies retest.

Trade On Bar Open

In this mode, the system will check for signals and places trades only at the open of a bar.

This is valid not only for opening the trade, but also for closing the trade on stop loss or profit target.

If the trade reaches its stop loss or profit target, it is **NOT closed immediately** at this level, but only at the open of the next bar!

If you'll use this mode you have to generate a special version of EA that will also trade only on bar open to achieve the same results.

Selected Timeframe only

together with Trade On Bar Open it is the fastest testing mode. It uses only the main timeframe to simulate the prices.

This results in a very fast backtesting with very good accuracy. However, for Stop or Limit orders the testing accuracy might not be sufficient and you should try more precise mode.

1 Minute data

slower testing mode, it uses minute data (if available) to simulate price changes during the testing.

Tick simulation

slow testing mode, it uses tick simulation within minute data to obtain more precise results. This is the most precise testing that can be achieved without real tick data.

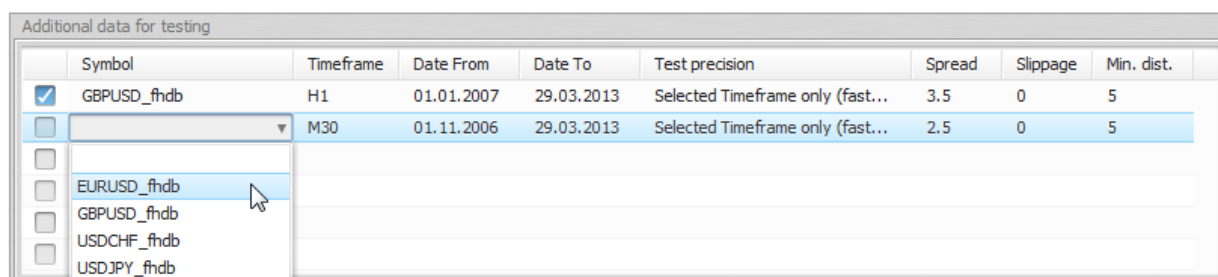
This testing mode is accurate enough for any type of orders.

Real Tick

this mode uses real tick data (if available) for exact price simulation. It is much slower than the other modes and it should be used for final verification of new strategies.

Additional data for testing

you can test your strategies automatically on multiple symbols/timeframes by specifying additional data.



To select additional data, choose the symbol in the combo box and set its timeframe, time range, spread, etc.

The strategy is tested on additional data only if the row is checked.

This way you can test your strategy simultaneously on multiple symbols, **or** on the same symbol and different timeframes or even on the same symbol and timeframe, and only a different spread, slippage or precision.

Viewing the results of additional data tests

If you select any additional data, you'll be able to check the results for each additional symbol/timeframe as well as result for whole **portfolio** (strategy tested on main data + all additional data summarized together) in the **Results** section.



If you'll use additional data for testing, you'll have also the opportunity to filter the resulting strategies by the performance of the whole portfolio in [Ranking Options Custom Conditions](#).

5.2.2 Building blocks

Building blocks are the core components that are combined together to create rules for every trading strategy.

Entry Rules Building Blocks

Entry building blocks can be divided into four main parts:

- price data - Open, High, Low, Close
- technical indicators -RSI, CCI, Momentum, etc.
- operators that are used for comparison and to combine the rules - <, >, and, or, etc.
- time constants - Hour, Minute of day, Day of week
- simple predefined rules - CCI > 0, Stochastic < 50 etc.

You can choose check each component that you want to use in the strategy, so you can select your favorite indicators,
or choose for example only price data + operators if you want to generate strategies based only on price.

Good practice

According to our experience, you can sometimes get better results if you don't check all the available components, but narrow your choice to a smaller group of indicators or price values.

You can find description of all [Entry and Exit nodes here](#).

5.2.3 Strategy options

These settings allow you to specify the properties of the generated strategies as well as testing conditions.

Market Sides

You can choose to generate strategies that trade only to one direction (Long or Short) or to both directions (which is standard).

You can also select that you want the entry or exit rules to be symmetrical. If they are symmetrical, then the rules for both directions are the same, only reversed.

An example of symmetrical rule s:

Go Long if CCI > 0

Go Short if CCI < 0

As an alternative, you can choose to use non-symmetrical rules, in this case the rules for Long and Short sides will be generated independently.

An example of non-symmetrical rules:

Go Long if CCI > 0

Go Short if RSI < 0 and Momentum < 100

This settings can be used for both entry and exit rules, for example you can have symmetrical entry rules, but non-symmetrical exit rules, so the strategy will effectively use (for example) different stop loss and profit target for Long and Short orders.

Trading Logic

Trading logic defines the behavior of the strategy and constraints of the backtester.

Exit at end of day/end of range

if selected, the strategy will close all position at the end of the day or end of the trading range (if defined). This way you'll have no position open overnight.

Maximum Trades Per Day

you can limit maximum trades the strategy takes per day

Maximum Total Trades

you can limit maximum total trades to a given number. It is usually good to use this limit and set it to some big number (1000-5000, depending on how many trades you expect during a tested period).

Replace Pending Orders

this is an important setting that has big influence on Stop and Limit orders. If set to true, the stop orders are replaced with updated version every time there is new signal and order is still pending. In the older versions of StrategyQuant (formerly named Genetic Builder) this setting was implicitly set to false and it was hidden.

It seems that setting it to true leads to better and more successful breakout strategies, and it is generally recommended.

Limit Signals to Time Range, Time Range From, Time Range To

this limits the hours the strategy is checking for entry signal to a given time range.

If used in combination with **Exit At End Of Range** then all open positions are closed at the end of the range.

If you don't check the **Exit At End Of Range** then the strategy will not open new trades outside the trading range, but the already opened positions will be not closed.

Exit At End Of Range, Exit At End Of Day, Exit On Friday

three settings that manage whether trades should be closed at the end of range, day or week.

Stop Loss & Profit Target Options

These settings allows you to specify whether Stop Loss and Profit Target should be mandatory in the strategy, and what is the minimum and maximum of the SL/PT values in pips.

You can also define the desired Risk Reward ratio.

Having defined SL/PT in the strategy is the simplest and many times the most effective approach.

If you unselect the mandatory SL/PT then the randomly generated strategy can (but doesn't have to) have fixed SL/PT. It is advisable to use different exit rule, for example exit after X bars or exit rule if you uncheck this setting, otherwise the strategy will have no way to exit the trade.

Lookback periods

Here you can set up the periods and coefficients used by the GB when generating the trading rules.

Maximum Period for Indicators

sets the maximum period that will be used for indicators. Usually we don't use periods bigger than 100-300 in the trading system (depending on the type of trading strategy), so it is advisable to set the desired maximum period value.

Maximum Period for Price Patterns

this is the maximum lookback period for price patterns (Open, High, Low, Close prices). It doesn't really make sense to set it to more than 10-20, because the older price usually has no relation with the current market movement.

Minimum and Maximum ATR Multiple

these are coefficients that are used in adaptive Stop Loss or Profit Target, if it is based on Average True Range value.

For example, setting Minimum to 0.5 and Maximum to 5 tells SQ that it can use minimum $0.5 * ATR$ as the SL or PT value and maximum $5 * ATR$ as the SL or PT value.

5.2.4 Genetic options

settings that affect genetic evolution.

An important thing is that there is not one ideal configuration. You can experiment with genetic evolution settings to obtain results that you want.

Genetic Options

Population size

is the number of different strategies in one population. It should be at least 50, the ideal value will be between 100 - 1000 or perhaps even more.

Max # of Generations

is the count of how many generations the evolution should be performed.

Mutation Probability

is the probability that a strategy will be mutated, it should be around 5-15%.

Crossover Probability

is the probability that the strategy is selected for crossover operation. Every generation there is a crossover of strategies performed during which the two parent strategies "mate" and produce two children.

The recommended value is 95%.

Selection type

Selection is an important part of an evolutionary algorithm. Without selection directing the algorithm towards fitter solutions there would be no progress.

Selection must favor fitter candidates over weaker candidates but beyond that there are no fixed rules. Furthermore, there is no one strategy that is best for all problems.

Some strategies result in fast convergence, others will tend to produce a more thorough exploration of the search space.

Sigma Scaling

attempts to moderate selection pressure over time so that it is not too strong in early generations and not too weak once the population has stabilized and fitness differences are smaller.

The Greek letter Sigma is used in statistics to denote standard deviation and that's what it means here too. The standard deviation of the population fitness is used to scale the fitness scores so that selection pressure is relatively constant over the lifetime of the evolutionary program.

Stochastic Sampling

is a variation of roulette wheel selection. It ensures that the observed selection frequencies of each individual are in line with the expected frequencies.

So if we have an individual that occupies 4.5% of the wheel and we select 100 individuals, we would expect on average for that individual to be selected between four and five times.

The individual will be selected either four times or five times, not three times, not zero times and not 100 times.

Tournament Selection

is among the most widely used selection strategies in evolutionary algorithms. It works well for a wide range of problems.

At its simplest tournament selection involves randomly picking two individuals from the population and staging a tournament to determine which one gets selected.

Initial Population Generation

for initial population you can choose two options:

Generate random population

Initial population will be generated randomly using the predefined settings and building blocks.

Decimation coefficient

decimation means that the system generates more strategies than needed for initial population, and then chooses only the best. For decimation coefficient is 1 it means there is no decimation.

For decimation coefficient = 2 system generates 2 x population size strategies and then chooses the best half for the initial population.

For decimation coefficient = 3 system generates 3 x population size strategies and then chooses the best third for the initial population.

Use systems from databank as initial population

existing strategies can be used as initial population. These strategies have to be loaded to the **Initial population** databank.

Initial Population Conditions

when generating initial population you can specify additional settings to filter bad strategies you don't want to use. For example you can configure it to dismiss (throw away) strategies where Net Profit is below zero or where number of trades is below certain level.

Evolution Management

here you can setup the repeated evolution. Top strategies from all the evolution runs are stored to Top strategies databank, so repeated evolution is a good way to let computer run for several hours or even days and keep it running continuous evolution.

Start again when finished

if checked, it will restart the evolution when it finished (reached maximum number of generations).

Restart when fitness stagnates for X generations

if checked, it will restart if fitness stagnates for given number of generation. Fitness is the strategy score. If it stagnates for 5-10 generations it is usually a sign that evolution reached its potential or went into a dead end, so it might be a good idea to just restart it.

5.2.5 Money management

Money management (or position sizing) specifies how many lots or shares are traded on each trade.

StrategyQuant contains three different money management options that can be used in the program and also later in real trading in MetaTrader.

Fixed size

strategy will trade with fixed number of lots. This is recommended setting when you generate new strategy, because it gives you clear overview of strategy real performance.

Fixed amount

strategy will risk a fixed amount of money for every trade.

This is basic money management without compounding. It can be used to test real performance of strategies where Stop Loss is based on volatility (ATR), or if you want to compare the performance of strategies with different Stop Loss.

Risk fixed % of account

advanced money management that is recommended for real trading. The strategy will risk a given % of equity on every trade.

This is simple, but very effective money management that will allow the strategy to increase number of lots as your account grows.

It is generally recommended to risk maximum 2-5% of account equity per one trade.

5.2.6 Robustness tests

Robustness tests are additional tests that can be applied to your strategy. The idea behind robustness testing is to verify how well the strategy will behave when there are small changes in inputs, history data or other components of the strategy.

Robust strategy should have good performance even when the input parameters are slightly changed, or when some orders are skipped.

StrategyQuant allows you to use Robustness tests in a form of Monte Carlo analysis. This means that the program will run a specified number of strategy tests with given random changes. As a result you'll receive a number of output results and output equities, one for each test.

You can then check these results to see how the strategy behaves when the inputs are changed, which helps you decide if you can trade such strategy.

Monte Carlo Simulation Settings

Testing the strategy with randomized input or history data provides only one alternative that can be compared to the original result.

In order to get more complete picture the robustness test can be repeated, until we receive bigger set of different results.

You can specify number of tests in the **Number of simulations** variable. The more simulations will be run, the better statistical significance of the results, but it comes at a price. Each test takes its time, so the more tests you run, the longer it will take to completely backtest the strategy.

The best value of tests is between 10 and 100.

Confidence level for reporting is simply a confidence level that is marked with a dark background in the Robustness analysis table (see below).

Usual confidence level to use is 95%.

Types of Robustness Tests

Below is an explanation of different types of tests you can use. If more than one option is checked, the tests are combined.

Randomize Trades Order

this is the simplest test, it randomly shuffles order of the trades. This doesn't change the resulting Net Profit, but it is very useful in examining different variations of Drawdown that can be a result of different order of trades.

Randomly Skip Trades

it will randomly skip trades with given probability. In real trading you can often miss a trade because of platform or Internet failure, or simply because you paused trading for some time.

This test will give you an idea how the equity curve might look like if some trades are randomly skipped.

Randomize Starting Bar

this will test the strategy behavior when the testing starts on a different starting bar. It is obvious that a good strategy cannot be sensitive to which bar you start the test.

Randomize Strategy Parameters

every strategy uses parameters, such as period of an indicator or constant that is used in comparison. This test checks the sensitivity of the strategy to a small change of parameter value.

Probability of change is a probability that any parameter changes its value. Max parameter change is the maximum percentage to which the parameter changes its value.

For example if you set Max parameter change to 10%, then a parameter with value 60 can be randomly changed to a range 54 - 66 (+- 10% of its original value of 60).

Randomize History Data

one very common case of curve fitting is when strategy is too dependent on history data. This option checks the behavior of the strategy to a change in history data.

The Probability of change sets for every bar how probable it is that open, high, low or close price will be changed. The Max price change is a percentage value of the change in relation to ATR (Average True Range).

So if for example close price is randomly chosen to be changed, ATR value is 10 pips, and Max price change is 20%, then the price can change by +- 2 pips

For information about how to use Robustness tests and how to interpret the results, check the [Robustness tests and analysis](#) section.

5.2.7 Ranking options

When the strategies are generated, every new strategy is backtested on a history data and the results of the backtest are then used to compute the *Fitness* (rank) of the strategy. Fitness is number from 0 to 1 and it should reflect the "quality" of the strategy according to the given criteria.

In this screen you can configure:

- how this *Fitness* value is computed (Strategy Selection Criteria criteria)
- how many top strategies are stored in the databank
- which strategies are stored to databank and which are thrown away (Custom conditions)

Databank Options

The best strategies found are continually stored into the **Databank**.

It is not possible to store every strategy (remember that StrategyQuant can create thousands of new strategies every hour) so we have to specify how many strategies should be stored in the databank, how they should be sorted to find out the best ones and which strategies should be thrown away.

Maximum Strategies to Store in Databank

simply the maximum number of strategies remembered in the program

Dismiss strategies that have 100% duplicate orders

if checked, strategies that have exactly the same orders as some strategy in databank will be dismissed. It is recommended to check this, if there is a strategy with duplicate orders, it will be likely the same strategy with redundant rules.

Custom Conditions

You can define rules to throw away strategies with bad properties. For example it usually makes sense to throw away strategies that result in loss or these that have too little number of trades.

Custom conditions allow you to specify your own custom rules that will be evaluated for every strategy.

If the strategy matches any of the rules, it will be dismissed.

This is useful to quickly dismissing strategies that result in loss or with too little number of trades.

There are separate values for In Sample and Out of Sample performance, and for Robustness Tests (if used) or Portfolio (if you used additional data).

Custom conditions - Dismiss strategies where:			
Use	What	<>=	Value
☒	Net Profit (IS)	<=	\$ 1000.0
☒	Number of trades (IS)	<=	250.0
☒	Return/DD Ratio (IS)	<=	2.5
☒	% Stagnation (IS)	>	30.0 %
☒	Return/DD Ratio (P)	<=	2.5

The abbreviations used in custom conditions are:

IS (In Sample) - result of In Sample part of data

OOS (Out of Sample) - result of In Sample part of data

RT (Robustness Tests) - result of robustness tests

P (Portfolio) - results for portfolio (if exists)

Strategy Selection Criteria

here you can choose the criteria you want to use to compute the total Fitness of the strategy.

You can choose from the predefined most used criteria, or you can build your own complex fitness function based on multiple criteria, each with a different weight.

Note that if you choose to combine too many criteria, they might "fight" against each other without achieving what you expected. **Profit Factor**

criterion to maximize Profit Factor of a strategy.

You can find explanation for every [ranking criteria here](#).

5.2.8 Parts to improve

Strategy consists of three parts - Entry rules, Order type and Exit rules. Each of these parts can be improved independently.

This way you can for example look for better exit rules, or generate a new rule for long entry while keeping the short entry rule from original strategy.

You have to check the part you want to improve. There is also a possibility to improve long/short parts independently.

Addition types:

add - adds new condition to the rule

replace - replace the whole rule with new condition

add or replace - randomly determines if to add or replace

Improvement process

Improvement means that StrategyQuant will make a copy the original strategy and randomly generate new conditions that will replace or add to the selected strategy part.

Then it'll test the new strategy variation to check if the "improvement" resulted in better performance.

Strategies that have better performance than original strategy are then stored to databank.

5.2.9 Parameters

this settings screen is available only for Optimizer. When you load a strategy to optimizer then you have to specify which parameters should be optimized in the Parameters screen.

The screenshot shows two overlapping dialog boxes. The top dialog, titled 'Optimization Results Options', contains a dropdown menu for 'Optimization method' set to 'Genetic Optimization' and a text field for 'Number of tests' set to '24'. The bottom dialog, titled 'Strategy Parameters', has two buttons: 'Load param values' and 'Save param values'. Below these buttons is a table with columns: '...', 'Name', 'Type', 'Original', 'Start', 'Stop', and 'Step'. Two rows are visible, both with checked checkboxes in the first column.

...	Name	Type	Original	Start	Stop	Step
☒	pEMA_1	Integer	10	5	20	5
☒	pEMA_2	Integer	20	10	60	10

Every parameter that you want to optimize has to be checked and you have to define its Start, Stop and Step values.

The optimization engine will try every value from start to stop making steps.

Original value is there for comparison with original strategy performance.

Number of tests

The value in the Number of tests shows you how many tests in total have to be performed.

If you'll choose to optimize more than one parameter, the optimization engine will test every combination of all parameters.

If you'll choose too many parameters to optimize, it can result to thousands or even millions of tests.

Optimization method

There are two optimization methods available:

Brute force - brute force methods tries every combination. This method

Genetic optimization - for every optimization with number of tests bigger than 500 or 1000 you should use Genetic Optimization method.

5.2.10 Success criteria

Walk-Forward Success Criteria are related to Walk-Forward optimization mode, they are automatically evaluated by the program and give us quick result whether the strategy passes or fails the test.

The screenshot shows the 'Optimization Engine' interface with the 'Success Criteria' tab selected. The 'Optimization Criteria' section contains the instruction 'Define criteria for passing the Walk Forward Analysis here.' and a 'Help' button. The 'Walk Forward Success Criteria' panel lists six criteria, each with a checkbox and a configurable value:

- Criterion 1: Overall Profitability**
☒ Fail if Total Profit < \$
- Criterion 2: Walk-forward Robustness**
☒ Fail if Walk-Forward Efficiency < %
- Criterion 3: Consistency of Profits**
☒ Fail if less than % of walk-forward runs were profitable
- Criterion 4: Distribution of Profits**
☒ Fail if any single run contributed more than % Total Net Profit
- Criterion 5: Maximum Drawdown %**
☒ Fail if maximum drawdown in any single run exceeded % of Initial Capital
- Criterion 6: Minimum trades**
☒ Fail if in-sample period produced less than trades

Walk-Forward optimization consists of periods of optimization part followed by run part.

1. Overall Profitability

overall profitability checks if the net profit result from optimization test is bigger than given value.

2. Walk-Forward Robustness

this criterion measures robustness of walk-forward runs. For example, if the optimization parts earned \$10.000 and run parts earned \$5.000, then the Walk-Forward Efficiency is exactly 50% (run parts earned 50% of what optimization parts).

3. Consistency of Profits

it checks how big percentage of run parts are profitable and compares it to given minimum percentage.

4. Distribution of Profits

this measures how big was the impact of every run to overall strategy profit. For example, if one period contributed to more than 50% of total strategy profit it means that the rest of periods performed quite poorly.

5. Maximum Drawdown

it checks if maximum percentage drawdown any run exceeded given maximum

6. Minimum trades

it checks that there is at least minimum number of trades produced in every run. If a run produces too little trades, it is less statistically significant.

5.2.11 Global options

Don't store list of trades in memory

If you check this option, all the strategies that are stored in databank will have their trades stored in temporary files. This greatly saves memory, because history of list of trades consumes the biggest part of the memory for every strategy.

If trades are stored in temporary files, they are loaded only when they are needed, for example when you'll switch to list of trades or to equity chart.

Don't store expired and replaced pending orders

if you check this options, expired and replaced pending orders are not saved to the strategy history. This is default setting, you should uncheck it only if you need to investigate something with the expired orders.

Strategy parameters

this tab allows you to set the global parameters behavior. Strategy contains two set of rules, one for long and one for short side. Usually they are symmetric.

Use the same parameters for long and short rules

Parameters used in these rules (for example periods for indicator) can be named by common names same for long and short rule, or separate for every direction.

Unless you want to optimize the strategy especially for long and short rule, you should leave it checked.

Add parameters for strategy options

this setting applies to Optimizer -> [Parameters](#) settings. When you'll optimize the strategy you can choose to display only strategy parameters, or also parameters for general trading options, such as time range for trade, etc.

Parameters without and with strategy options:

Strategy Parameters						
Load param values Save param values						
...	Name	Type	Original	Start	Stop	Step
☐	pEMA_1	Integer	10	10	10	0
☐	pEMA_2	Integer	20	20	20	0

Strategy Parameters						
Load param values Save param values						
...	Name	Type	Original	Start	Stop	Step
☐	pEMA_1	Integer	10	10	10	0
☐	pEMA_2	Integer	20	20	20	0
☐	Maximum Trades Per Day	Integer	0	0	0	0
☐	Maximum Total Trades	Integer	0	1000	0	0
☐	Replace Pending Orders	Boolean	False	False	False	False
☐	Limit Signals to Time Range	Boolean	False	False	False	False
☐	Time Range From	Time	00:00	00:00	00:00	00:00
☐	Time Range To	Time	00:00	16:00	00:00	00:00
☐	Exit At End Of Range	Boolean	False	False	False	False

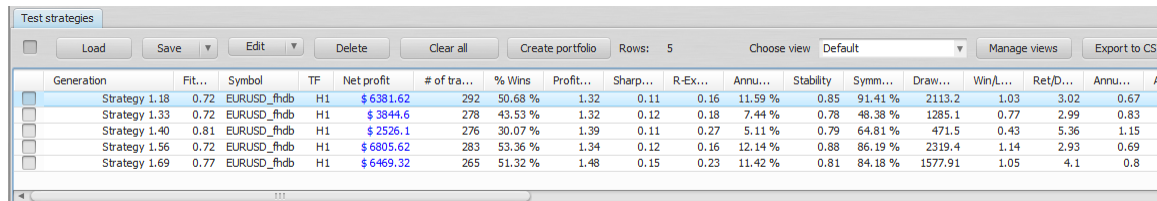
HTML reports

possibility to customize HTML reports produced by the program. You can customize which charts you want to see in the strategy performance report.

5.3 Results

5.3.1 Databanks

Databank is a storage place where the generated/tested strategies are stored with their results.



The screenshot shows a window titled 'Test strategies' with a table of strategy results. The table has columns for Generation, Fit..., Symbol, TF, Net profit, # of tra..., % Wins, Profit..., Sharp..., R-Ex..., Annu..., Stability, Symm..., Draw..., Win/L..., Ret/D..., Annu..., and A. The first four rows are visible, showing strategies 1.18, 1.33, 1.40, and 1.56. Strategy 1.18 has a net profit of \$6381.62, while the others are negative.

Generation	Fit...	Symbol	TF	Net profit	# of tra...	% Wins	Profit...	Sharp...	R-Ex...	Annu...	Stability	Symm...	Draw...	Win/L...	Ret/D...	Annu...	A
Strategy 1.18	0.72	EURUSD_fhdb	H1	\$6381.62	292	50.68 %	1.32	0.11	0.16	11.99 %	0.85	91.41 %	2113.2	1.03	3.02	0.67	
Strategy 1.33	0.72	EURUSD_fhdb	H1	\$3844.6	278	43.53 %	1.32	0.12	0.18	7.44 %	0.78	48.38 %	1285.1	0.77	2.99	0.83	\$
Strategy 1.40	0.81	EURUSD_fhdb	H1	\$2526.1	276	30.07 %	1.39	0.11	0.27	5.11 %	0.79	64.81 %	471.5	0.43	5.36	1.15	\$
Strategy 1.56	0.72	EURUSD_fhdb	H1	\$6805.62	283	53.36 %	1.34	0.12	0.16	12.14 %	0.88	86.19 %	2319.4	1.14	2.93	0.69	\$

Each mode (Build/Test/Improve/Optimize) has its own independent databank table.

For memory reasons Databank cannot keep unlimited number of strategies, instead it stores the selected number of top strategies, for example top 100 or top 1000 strategies.

The configuration how many strategies to store in the Databank and how to sort them can be set in the Settings -> [Ranking Options](#)

Databank functions

Every strategy result in the Databank can be viewed in the Results screen by using double-click.

Load

this button is available only in **Retest** databank, it loads Strategy projects (.str files) so they can be retested.

Save (as Strategy/Portfolio/HTML report/Source code)

this way you can save your strategy, you should always save good strategies to a Strategy Project (.str) so that you can work with them later.

Edit (Parameters/Trading rules)

here you can manually edit the selected strategy - either the values of its parameters or the trading rules.

Delete/Clear all

this will delete selected or all strategies from the list

Create portfolio

it will combine selected strategies to a portfolio - a special report that combines trading results of multiple strategies. You can then check/analyze the results of whole portfolio of strategies.

Views

Each row in the table represents one strategy plus, the results of the strategy backtest - number of trades, total profit/loss, Fitness score, etc. are visible in the table columns.

Results are divided into In Sample and Out of Sample parts.

All the columns in the databank are customizable, you can choose which columns you want to see using views.

Export to CSV

you can also export the whole table content to a CSV file

5.3.2 Overview

Overview screen displays various statistics computed from the backtest data.

5.3.3 List of trades

Contains complete list of trades generated by the backtest.

If you backtested your strategy on multiple symbols/timeframes, you can switch between the results or check the list of trades for whole portfolio.

5.3.4 Equity chart

Displays equity chart of the strategy.

Stagnation marker shows the maximum stagnation period - this is the longest period that it took strategy to make new high on the equity.

In case of portfolio, it displays charts for every symbol/timeframe tested, plus equity of complete portfolio.

5.3.5 Trade analysis

This screen displays yearly performance of the strategy plus configurable small charts with trades per hour, day of week, day of month, etc.

5.3.6 Robustness analysis

This screen is empty if you didn't configure any Robustness Tests.

Otherwise it contains equity chart for every robustness test simulation and table of important confidence levels computed using Monte Carlo analysis of simulations.

For information about how to use Robustness tests and how to interpret the results, check the [Robustness tests and analysis](#) section.

5.3.7 Strategy settings

This screen displays current settings of the program versus settings that were used when the strategy was last time backtested/generated.

The differences are highlighted with **red color**.

To quickly load settings from the strategy you can use the button **Apply Strategy Settings**.

This will load settings from the strategy and applies it to the selected program section.

Overview

List of trades

Equity chart

Trade analysis

Robustness analysis

Strategy settings

Source code

Refresh Current Settings

Apply Strategy Settings

from/to task

Retest strategies

Name	Strategy last test	Current settings	Name	Strategy last test	Current settings
Build Goals					
Market Sides					
Type	Both Long & Short	Both Long & Short			
Long / Short Entry Rules symmetry	True	True	Long / Short Exit Rules symmetry	True	True
Strategy Options					
Trading Logic					
Exit at end of day / end of range	False	False	Limit max trades per day to	False	False
Limit max total trades to	10000	10000	Limit trading to hours	False	False
Close also pending Stop/Limit orders	True	True	Enter only once per bar	True	True
			Check signals only on bar open	True	True
Stop Loss & Profit Target options					
Stop Loss is Required	True	True	Profit Target is Required	True	True
Minimum SL/PT in pips	30	30	Maximum SL/PT in pip	500	300
Lookback Periods					
Maximum Period for Indicators	100	100	Maximum Period for Price Patterns	20	20
Minimum ATR Multiple	0.5	0.5	Maximum ATR Multiple	5.0	5.0
Data					
Symbol	GBPUSD_mt4	EURUSD_a	Timeframe	H1	H1
Start Date	01.01.2000 00:00:00	01.01.2006 00:00:00	End Date	28.10.2011 00:00:00	30.05.2012 23:59:59
OOS Start Date	28.10.2011 00:00:00	30.05.2012 23:59:59	OOS End Date	28.10.2011 00:00:00	30.05.2012 23:59:59
Spread	3.5	2.5	Market slippage	0.0	0.0
Min distance from price	5	5	Test precision	Tick simulation (slow...)	1 Minute data (slow)
Money Management					
Type	Fixed size	Fixed size	Capital size	10000	10000
Details	0.1 lots	0.1 lots			

5.3.8 Source code

Source code tab is where you'll get the result of the program. For every selected strategy it generates selected source code.

Source Code Type

There are three types of source code that can be generated:

Pseudo Code

human readable pseudo code of the strategy. You can see the strategy logic and you can use it for manual trading.

MetaTrader4 Expert Advisor

the EA code for MetaTrader. You can save the EA code to MetaTrader/experts directory and you'll be able to run your new EA in MetaTrader.

MT4 Special EA - Trade on Bar Open

special version of MetaTrader4 EA - it checks for signals and **places trades only at the open of a bar**. This is valid not only for opening the trade, but also for closing the trade on stop loss or profit target. If the trade reaches its stop loss or profit target, it is **NOT closed immediately** at this level, but at the open of the next bar!

EA generated this way will be 100% compatible with the strategy tested in GB using the **Trade on Bar Open** test precision.

Put values to parameters

by checking this checkbox you can generate code where all the number constants like indicator periods, comparison constants etc.

will be parameterized - a parameter will be added for every constant.

This will allow you to optimize your EA in MetaTrader, or test it with different parameter values.

6 Advanced functionality

6.1 Robustness tests and analysis

Robustness tests are additional tests that can be applied to your strategy. The idea behind robustness testing is to verify how well the strategy will perform when there are small changes in inputs, history data or other components of the strategy.

Robust strategy should have good performance even when the input parameters are slightly changed, or when some orders are skipped.

Types of Robustness Tests

Below is an explanation of different types of tests you can use. If more than one option is checked, the tests are combined.

Randomize Trades Order

this is the simplest test, it randomly shuffles order of the trades. This doesn't change the resulting Net Profit, but it is very useful in examining different variations of Drawdown that can be a result of different order of trades.

Randomly Skip Trades

it will randomly skip trades with given probability. In real trading you can often miss a trade because of platform or Internet failure, or simply because you paused trading for some time. This test will give you an idea how the equity curve might look like if some trades are randomly skipped.

Randomize Starting Bar

this will test the strategy behavior when the testing starts on a different starting bar. It is obvious that a good strategy cannot be sensitive to which bar you start the test.

Randomize Strategy Parameters

every strategy uses parameters, such as period of an indicator or constant that is used in comparison. This test checks the sensitivity of the strategy to a small change of parameter value.

Probability of change is a probability that any parameter changes its value. Max parameter change is the maximum percentage to which the parameter changes its value.

For example if you set Max parameter change to 10%, then a parameter with value 60 can be randomly changed to a range 54 - 66 (+- 10% of its original value of 60).

Randomize History Data

one very common case of curve fitting is when strategy is too dependent on history data. This option checks the behavior of the strategy to a change in history data.

The Probability of change sets for every bar how probable it is that open, high, low or close price will be changed. The Max price change is a percentage value of the change in relation to ATR (Average True Range).

So if for example close price is randomly chosen to be changed, ATR value is 10 pips, and Max price change is 20%, then the price can change by +- 2 pips

Monte Carlo Simulation Settings

Testing the strategy with randomized input or history data provides only one alternative that can be compared to the original result.

In order to get more complete picture the robustness test can be repeated, until we receive bigger set of different results.

You can specify number of tests in the **Number of simulations** variable. The more simulations will be run, the better statistical significance of the results, but it comes at a price. Each test takes its time, so the more tests you run, the longer it will take to completely backtest the strategy.

The best value of tests is between 10 and 100.

Confidence level for reporting is simply a confidence level that is marked with a dark background in the Robustness analysis table (see below).

Usual confidence level to use is 95%.

What is Monte Carlo Simulation/Analysis?

Monte Carlo simulation is a problem solving technique used to approximate the probability of certain outcomes by running multiple trial runs, called simulations, using random variables.

By running multiple simulations with random robustness tests we can perform the Monte Carlo analysis on the results.

Let's run a simple simulation to test how the strategy is affected by randomly missed trades. We will run 10 simulations, and there will be 10% probability that any trade is skipped.

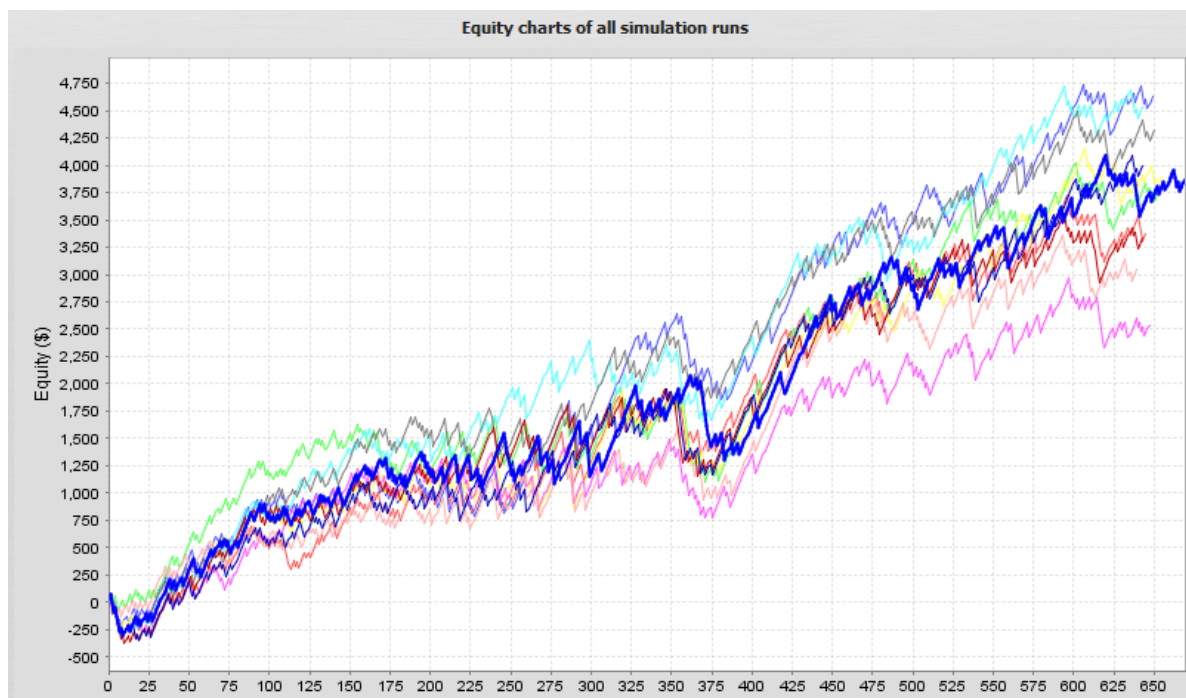
The image shows two overlapping dialog boxes from the StrategyQuant software. The top dialog, titled "Monte Carlo Simulation Settings", has a checked checkbox for "Use Robustness Testing Monte Carlo Analysis". Below this, "Number of simulations" is set to 10 and "Confidence level for reporting" is set to 95%. The bottom dialog, titled "Robustness Testing Methods", contains several options: "Randomize Trades Order" is unchecked; "Randomly Skip Trades" is checked with a "Skip trade probability" of 10%; "Randomize Starting Bar" is unchecked with a "Max starting bar change" of 100; "Randomize Strategy Parameters" is unchecked with "Probability of change" at 20% and "Max parameter change" at 20%; and "Randomize History Data" is unchecked with "Probability of change" at 50% and "Max price change" at 20% of ATR.

Now when you run build or retest, these robustness tests will be performed for every strategy as a part of the building of testing process.

Robustness analysis results

Robustness analysis shows us 10 different curves, plus the equity curve for original strategy (without any trade skipped).

You can see that depending on the trades skipped, the results of the strategy can differ significantly.



If you'll look at the table on the left of the chart, you'll see the Monte Carlo analysis of the results.

Confidence levels for robustness tests

The levels below show with what confidence you can expect the displayed strategy results. Computed results are based on Monte Carlo simulation of multiple runs made during robustness tests.

Results in the table are for complete data (In Sample + Out of Sample).

Conf	Net Profit	Max DD %	Ret/DD	R Exp	AR %
Original	\$ 3867.98	6.32 %	5.07	0.06 R	5.28 %
50 %	\$ 3899.97	6.22 %	5.08	0.07 R	5.32 %
60 %	\$ 3738.37	6.23 %	5.07	0.06 R	5.13 %
70 %	\$ 3377.88	6.46 %	4.64	0.06 R	4.72 %
80 %	\$ 3335.38	6.58 %	4.29	0.06 R	4.66 %
90 %	\$ 3046.88	6.67 %	4.18	0.05 R	4.33 %
91 %	\$ 2539.6	7.27 %	3.55	0.04 R	3.71 %
92 %	\$ 2539.6	7.27 %	3.55	0.04 R	3.71 %
93 %	\$ 2539.6	7.27 %	3.55	0.04 R	3.71 %
94 %	\$ 2539.6	7.27 %	3.55	0.04 R	3.71 %
95 %	\$ 2539.6	7.27 %	3.55	0.04 R	3.71 %
96 %	\$ 2539.6	7.27 %	3.55	0.04 R	3.71 %
97 %	\$ 2539.6	7.27 %	3.55	0.04 R	3.71 %
98 %	\$ 2539.6	7.27 %	3.55	0.04 R	3.71 %
99 %	\$ 2539.6	7.27 %	3.55	0.04 R	3.71 %
100 %	\$ 2539.6	7.27 %	3.55	0.04 R	3.71 %

What do these values mean?

The first row displays values of Net Profit, Maximum % Drawdown etc. of original strategy for comparison.

The rest of the rows display values at different confidence levels.

These numbers are a result of Monte Carlo analysis applied on our 10 random simulations.

For example, values at 90% confidence level mean that there is 10% chance that Net Profit, Drawdown will be worse than the confidence level values.

Values at 95% confidence level mean that there is only a 5% chance that Net Profit, Drawdown, etc. will be worse than these values.

In this test we randomly skipped trades. Monte Carlo analysis shows us that by skipping 10% of random trades our Net Profit can decrease from \$3800 to \$2500, and Maximum Drawdown can increase from 6.32% to 7.27%, and there is only 5% probability that our values will be worse than

This means that even if our strategy has Net Profit of \$ 3867, Monte Carlo analysis shows us that **by skipping just 10% of the trades the Net Profit can decrease**, and there is 5% chance that Net Profit will be lower than \$ 2539. By looking at the higher confidence levels we can see that none of our tests had worse results than \$ 2539.

Because robustness tests are generated randomly, equity charts and values in the table will slightly differ every time you retest the strategy.

Also, the more simulations you'll run, the bigger statistical significance of this test.

Robustness Test Example 2 - Randomizing strategy parameters and history data

In second example we'll run robustness tests that will randomize both input data and history price. This time we'll run 30 simulations to get more statistical data.

Monte Carlo Simulation Settings

☒ **Use Robustness Testing Monte Carlo Analysis**

Number of simulations

30

Confidence level for reporting

95

%

Robustness Testing Methods

☐ **Randomize Trades Order**

☐ **Randomly Skip Trades**

Skip trade probability

10

%

☐ **Randomize Starting Bar**

Max starting bar change

100

☒ **Randomize Strategy Parameters**

Probability of change

20

%

Max parameter change

10

%

☒ **Randomize History Data**

Probability of change

20

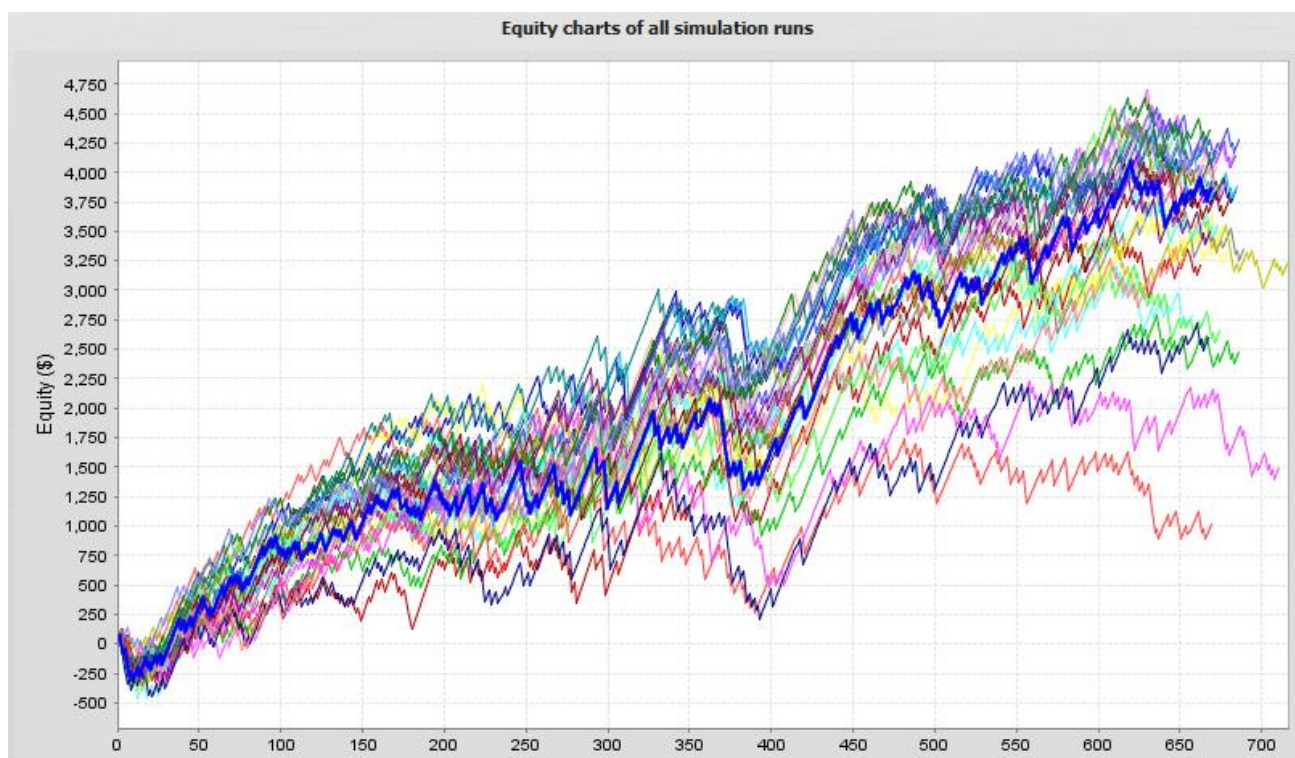
%

Max price change

20

% of ATR

When the backtest finishes we can see that the chart shows a bigger variety of different equity curves.



The look at the confidence levels table shows us that at 95% confidence level the drawdown doubles and Net Profit is less than half of the profit of original strategy, however the strategy is still profitable so it seems to be robust enough.

Confidence levels for robustness tests

The levels below show with what confidence you can expect the displayed strategy results. Computed results are based on Monte Carlo simulation of multiple runs made during robustness tests.

Results in the table are for complete data (In Sample + Out of Sample).

Conf	Net Profit	Max DD %	Ret/DD	R Exp	AR %
Original	\$ 3867.98	6.32 %	5.07	0.06 R	5.28 %
50 %	\$ 3844.46	5.71 %	5.1	0.06 R	5.25 %
60 %	\$ 3534.98	6.27 %	4.89	0.06 R	4.9 %
70 %	\$ 3331.86	6.34 %	4.63	0.06 R	4.67 %
80 %	\$ 3207.79	7.46 %	3.5	0.05 R	4.53 %
90 %	\$ 2598.39	8.44 %	2.79	0.04 R	3.81 %
91 %	\$ 2480.88	9.46 %	1.86	0.04 R	3.63 %
92 %	\$ 2480.88	9.46 %	1.86	0.04 R	3.63 %
93 %	\$ 2480.88	9.46 %	1.86	0.04 R	3.63 %
94 %	\$ 1496.41	12.03 %	0.95	0.02 R	2.36 %
95 %	\$ 1496.41	12.03 %	0.95	0.02 R	2.36 %
96 %	\$ 1496.41	12.03 %	0.95	0.02 R	2.36 %
97 %	\$ 1012.98	13.07 %	0.94	0.01 R	1.68 %
98 %	\$ 1012.98	13.07 %	0.94	0.01 R	1.68 %
99 %	\$ 1012.98	13.07 %	0.94	0.01 R	1.68 %
100 %	\$ 1012.98	13.07 %	0.94	0.01 R	1.68 %

6.2 Optimization

6.2.1 Simple optimization

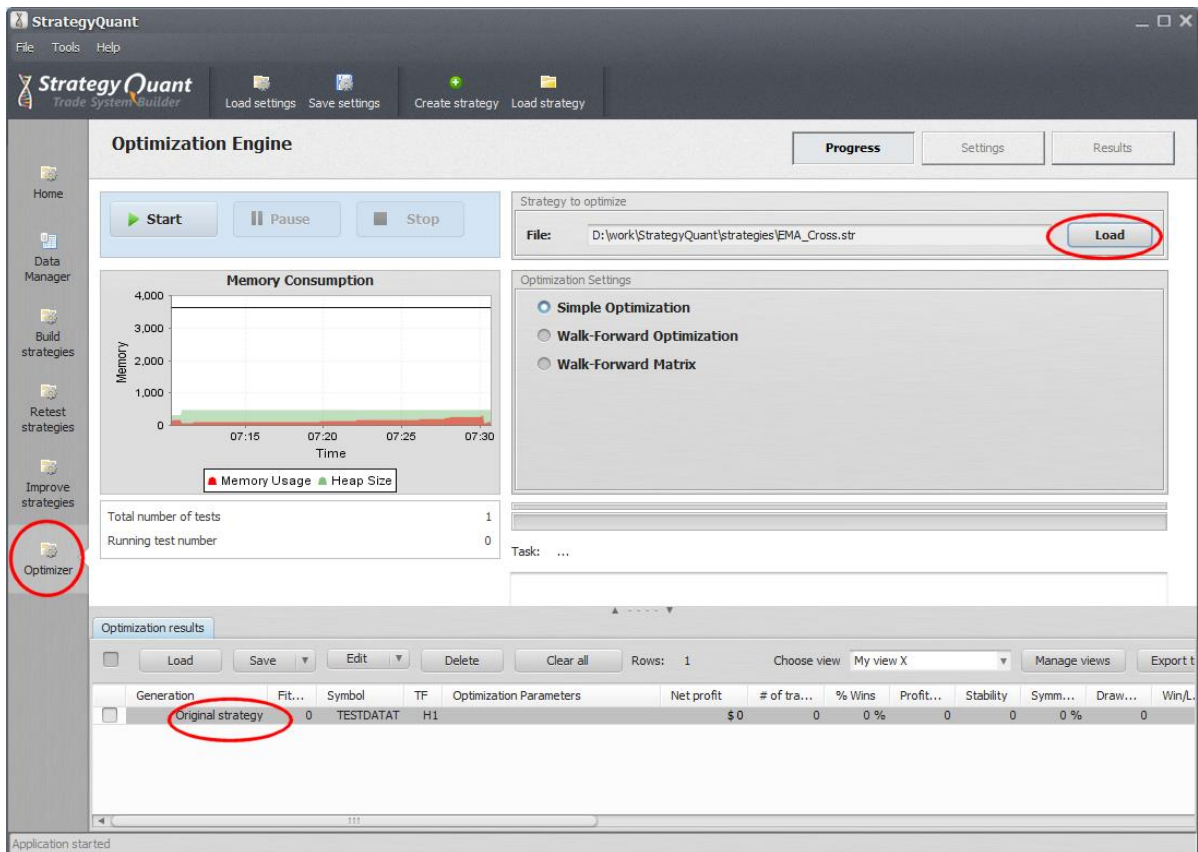
The idea behind an optimization is simple. First you have to have a trading system, this may be a simple moving average crossover for example.

In almost every system there are some parameters (indicator periods, comparative constants, etc.) that decide how given system behave. The optimization means to test the system with different parameter values to find the optimal values of these parameters (giving highest profit or best Return/DD ratio).

Optimization example

Step 1: Loading a strategy for optimization

First of all, you have to switch to Optimizer window and load the strategy you want to optimize.



For this example we'll use simple EMA_Cross strategy that goes long when faster EMA crosses above slower EMA, and go short when faster EMA crosses below slower EMA. After you loaded the strategy, it is added also as Original strategy to the Optimization results databank.

You can double click on the Original strategy and then go to Results -> Source code to see its rules.

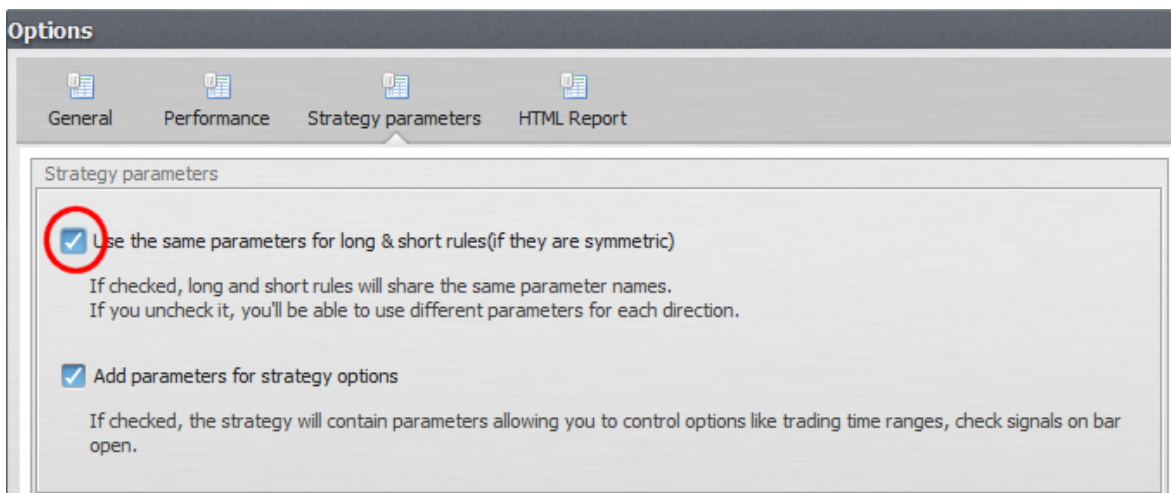


Make sure you check the **Put values to parameters** checkbox so that you see that variables pLongEMA_1, pLongEMA_2, pShortEMA_1, pShortEMA_1 are used to store indicator parameters. In our optimization we'll try to find optimal values of these parameters.

There's still one small problem. We can see that the strategy uses different parameters for long and for short direction.

We can use it like this if we want to find optimal values independently for long and short side, but for our example we'd like to use the same parameter for long and short side.

We can do it in program **Tools -> Options -> Strategy parameters**.



If you check the first checkbox, it will use the same parameters for long and short direction (providing the rules are the same).

Click OK to store the settings and refresh the source code.

```

=====
== Strategy Parameters
=====

pEMA_1 = 10;
pEMA_2 = 20;

=====

== Entry conditions
=====

LongEntryCondition = (EMA(pEMA_1) Crosses Above EMA(pEMA_2))
ShortEntryCondition = (EMA(pEMA_1) Crosses Below EMA(pEMA_2))

=====

== Entry orders
=====

-- Long entry
if LongEntryCondition is true {
    Reverse existing order (if any) and Buy on open at Market;
}

-- Short entry
if ShortEntryCondition is true {
    Reverse existing order (if any) and Sell on open at Market;
}

```

Now you can see that the strategy now contains only parameters pEMA_1, pEMA_2 that are used for both directions.

Step 2: Setting optimization values

To set up values that will be optimized we have to go to Settings -> Parameters

Optimization Engine

Progress

Settings

Data

Parameters

Success Criteria

Money Management

Optimization Parameters

Choose which parameters of the strategy will be optimized.

?

Help

Optimization Results Options

Optimization method

Genetic Optimization

Number of tests:

24

Optimization Results Options

Maximum strategies to store in results

100

Strategy Parameters

Load param values

Save param values

	Name	Type	Original	Start	Stop	Step
☒	pEMA_1	Integer	10	5	20	5
☒	pEMA_2	Integer	20	10	60	10

Strategy Selection Criteria - Determine Best Strategies by

☐ Net Profit (Return)
 ☒ Return / Drawdown ratio
 ☐ Strategy Quality Number (Van Tharp)
 ☐ Strategy Quality Number score (Van Tharp + MF)
 ☐ Expectancy (Van Tharp)

Here you can see the list of all strategy parameters that are available for optimization. Optimization simply means trying different values of input parameters.

For every parameter you want to optimize you have to **check** the line of the parameter and choose **Start**, **Stop** and **Step** values. The optimizer will iterate the value from Start to Stop, taking Steps.

Original value is also configurable, it will be used to retest the original strategy. You can use this value to compare performance of new results with the "original" settings.

The **Number of tests** value shows us how many tests have to be performed to test all the combinations of the values.

Note!

It is possible that your parameters table will contain much more parameters, it could look like this:

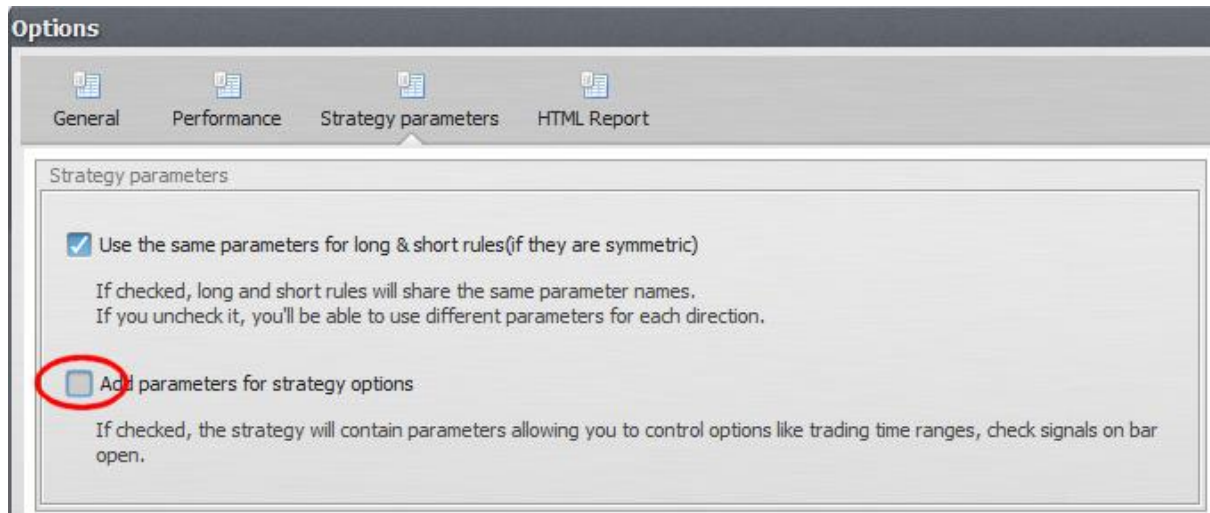
The screenshot displays two windows from the StrategyQuant software. The top window, titled "Optimization Results Options", shows the "Optimization method" set to "Genetic Optimization" and the "Number of tests" set to 24. The bottom window, titled "Strategy Parameters", contains a table of parameters for optimization. The table has columns for a selection checkbox, Name, Type, Original, Start, Stop, and Step. Parameters include pEMA_1, pEMA_2, Maximum Trades Per..., Maximum Total Trades, Replace Pending Ord..., Limit Signals to Time..., Time Range From, Time Range To, Exit At End Of Range, Exit At End Of Day, and Exit On Friday. Checkboxes for pEMA_1 and pEMA_2 are checked, while others are unchecked.

...	Name	Type	Original	Start	Stop	Step
☒	pEMA_1	Integer	10	5	20	5
☒	pEMA_2	Integer	20	10	60	10
☐	Maximum Trades Per...	Integer	0	0	0	0
☐	Maximum Total Trades	Integer	0	1000	0	0
☐	Replace Pending Ord...	Boolean	False	False	False	False
☐	Limit Signals to Time ...	Boolean	False	False	False	False
☐	Time Range From	Time	00:00	00:00	00:00	00:00
☐	Time Range To	Time	00:00	16:00	00:00	00:00
☐	Exit At End Of Range	Boolean	False	False	False	False
☐	Exit At End Of Day	Boolean	False	False	False	False
☐	Exit On Friday	Boolean	False	False	False	False

This is another powerful feature of StrategyQuant. It allows you to optimize not only strategy parameters, but also other trading options, such as how many trades to take per day, or what should be the time range for trading.

These settings are normally a part of Strategy Options, but you can also optimize their values.

If you don't want to use them and see them in Parameters table, go once again to **Tools -> Options -> Strategy parameters** and uncheck the checkbox for **Add parameters for strategy options**.



The last thing we have to configure is the data that will be used for testing. We can choose EURUSD on H4 timeframe for this example.

Step 3: Running the optimization

Now we are ready to run the optimization. We'll go back to Progress screen and click on **Start** button. The optimization engine will test all the possible combinations of selected input parameters and stores the results for each combination to the databank on the bottom.

The screenshot shows the 'Optimization Engine' window. At the top, there are buttons for 'Progress', 'Settings', and 'Results'. Below these, there are buttons for 'Start', 'Pause', and 'Stop'. The 'Start' button is circled in red. To the right of these buttons is a 'Strategy to optimize' section with a file path and a 'Load' button. Below that is the 'Optimization Settings' section with three radio buttons: 'Simple Optimization' (selected), 'Walk-Forward Optimization', and 'Walk-Forward Matrix'. On the left, there is a 'Memory Consumption' graph showing 'Memory Usage' (red) and 'Heap Size' (green) over time. Below the graph, it says 'Total number of tests: 24' and 'Running test number: 0'. At the bottom, there is a 'Task' status bar showing 'DONE in 1.957 s.' and a list of test results. The 'Optimization results' table at the bottom has columns: Generation, Fit..., Symbol, TF, Optimization Parameters, Net p..., # of tra..., % Wins, Profit..., Stability, Symm..., Draw..., and Win. The row for 'Optimized 0.14' is circled in red, showing a net profit of \$2506.7.

Generation	Fit...	Symbol	TF	Optimization Parameters	Net p...	# of tra...	% Wins	Profit...	Stability	Symm...	Draw...	Win
Optimized 0.0	0.61	EURUSD_fhdb	H4	pEMA_1=5.0, pEMA_2=10.0	\$ 3953.95	535	33.08 %	1.17	0.74	49.09 %	2483.27	
Optimized 0.4	0.5	EURUSD_fhdb	H4	pEMA_1=5.0, pEMA_2=20.0	\$ 3557.12	377	29.71 %	1.19	0.54	45.8 %	4522.69	
Optimized 0.23	0.57	EURUSD_fhdb	H4	pEMA_1=20.0, pEMA_2=60.0	\$ 3337.11	117	30.77 %	1.33	0.74	45.87 %	2677.59	
Optimized 0.20	0.52	EURUSD_fhdb	H4	pEMA_1=5.0, pEMA_2=60.0	\$ 2993.53	223	26.01 %	1.21	0.65	39.73 %	3227.59	
Original strategy	0.48	EURUSD_fhdb	H4		\$ 2760.33	273	30.77 %	1.17	0.56	36.26 %	3849.09	
Optimized 0.14	0.59	EURUSD_fhdb	H4	pEMA_1=15.0, pEMA_2=40.0	\$ 2506.7	165	30.91 %	1.19	0.64	31.38 %	1824.7	

We can sort the databank by Net Profit and we can see that the best input values in terms of maximum profit are pEMA_1= 5 and pEMA_2 = 10.

We can also see the results of original strategy (gray background) that used pEMA_1= 10 and pEMA_2 = 20

Interpreting the results

Now we've got input parameters that were optimized for our given symbol and timeframe.

What we've really done is we found out what worked best in the past. We have to be very careful because the parameters might be ideal for the history data, but there is no guarantee that what worked best on history data will work also in the future.

It is called curve fitting - usually the more parameters the strategy has, the bigger is the danger of curve fitting.

There are two approaches to curve fitting:

- make sure the strategy is robust and not optimizing its values at all
- make sure the strategy benefits from periodic reoptimization

So the question is - will periodical reoptimization improve the results of my strategy?

If yes, how often should I do it?

StrategyQuant can give you answers to these questions using another of its advanced functionality - [Walk-Forward Optimization](#) and [Walk-Forward Matrix](#).

6.2.2 Walk-Forward Optimization

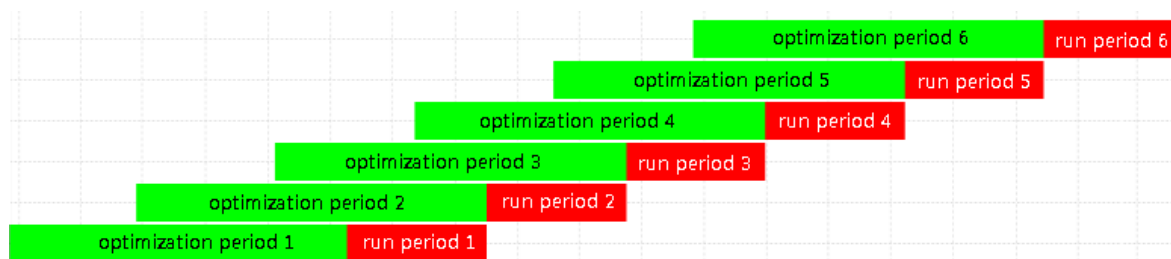
Walk-Forward optimization is generally a set of simple optimizations followed by trading using the best parameters.

It is a technique in which you optimize the parameter values on a past segment of market data, then verify the performance of the system by testing it forward in time on data following the optimization segment, and the process can be repeated over subsequent time segments.

How Walk-Forward optimization works

in walk-forward optimization, the data are divided into a configurable number of periods (6 in this example).

Each period consists of optimization part and run part.



The program starts with optimization period 1. It will run the simple optimization on optimization period 1 to find the best parameter values.

These parameter values are then applied to run period 1 - strategy is tested by trading with the optimized parameters on the run period.

At the end of run period 1, the system again runs simple optimization on a part of data marked as optimization period 2.

It finds the best set of parameter values and they are again used for trading in run period 2.

This continues until the period 6, which is also the end of history data used in test.

Walk-Forward optimization simulates how you'd work with the strategy during real trading - you'd optimize it on some historical data and then trade it with the optimal values.

After some time you'd want to reoptimize it and let it trade again.

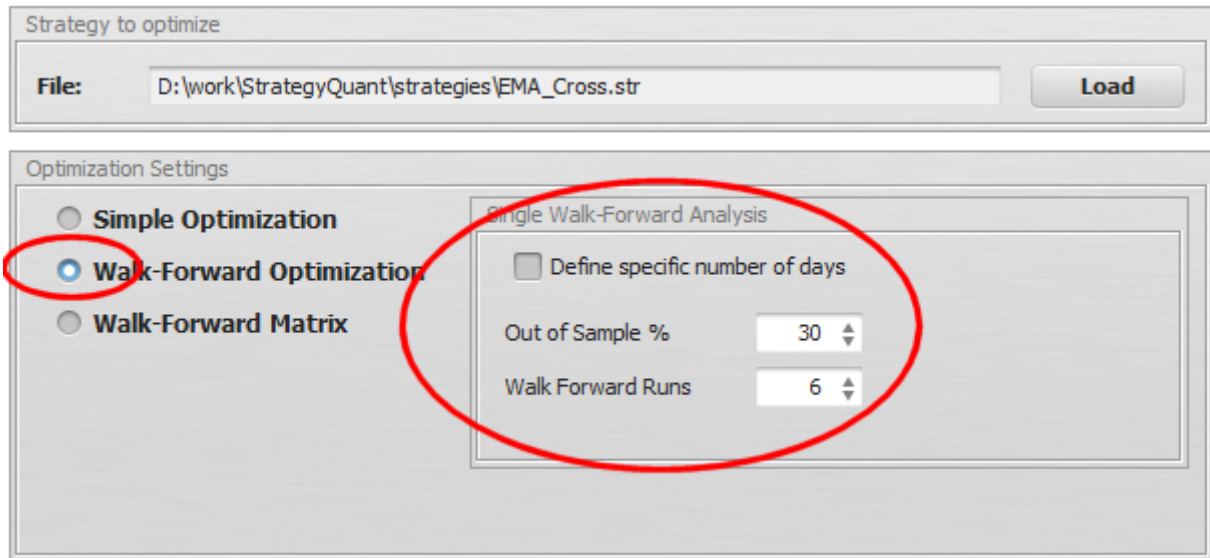
Analysis of walk-forward results can tell you if the strategy is a good candidate for periodic reoptimization.

Walk-Forward optimization example

Step 1: Configuring walk-forward runs

we'll follow with our [Simple optimization example](#), we have to load our strategy and choose the parameter values to be optimized,

only now we'll choose **Walk-Forward Optimization** option in the Progress screen.



For this option we have to specify also the walk-forward settings.

Out of Sample %

this means how much of the whole period is left for run. If we'll set it to 30 %, it means that in each period 70% of the data are used for optimization, and 30% will be used for trading using the optimized values.

Walk Forward Runs

this means how many optimization runs there will be, which means how many times we'll reoptimize the strategy.

It is also possible to specify the optimization (In Sample) and run (Out of Sample) periods by exact days, you can do it by checking **Define specific number of days**.

Step 2: Configuring optimization criteria

We can (optionally) set up also optimization success criteria - they are automatically evaluated by the program and they'll give us an exact result if our strategy passed or failed the Walk-Forward Optimization.

We'll set up our criteria as in the picture below.

Optimization Engine

Progress

Settings

Data

Parameters

Success Criteria

Money Management

Optimization Criteria

Define criteria for passing the Walk Forward Analysis here.

Help

Walk Forward Success Criteria

Criterion 1: Overall Profitability

☒ Fail if Total Profit < \$ 0

Criterion 2: Walk-forward Robustness

☒ Fail if Walk-Forward Efficiency < 50 %

Criterion 3: Consistency of Profits

☒ Fail if less than 60 % of walk-forward runs were profitable

Criterion 4: Distribution of Profits

☒ Fail if any single run contributed more than 50 % Total Net Profit

Criterion 5: Maximum Drawdown %

☒ Fail if maximum drawdown in any single run exceeded 25 % of Initial Capital

Criterion 6: Minimum trades

☒ Fail if in-sample period produced less than 30 trades

To find more information about what each criterion means, check the [Success Criteria](#) page.

Now we can go back to progress panel and click on the **Start** button.

Step 3: Checking the results

Walk-Forward optimization takes longer than a simple one, because there are 6 (or more) optimization steps instead of just one.

When it finishes, we'll see that we have only two results in the databank - Original strategy and Walk-Forward result.

Walk-Forward results													
☐	Load	Save	▼	Delete	Clear all	Rows: 2							
Generation	Fit...	Symbol	TF	Net profit	# of tra...	% Wins	Profit...	Sharp...	R-Ex...	Annu...	Stability	Symm...	Dra
☐ Original strategy	0.48	EURUSD_fndb	H4	\$ 2760.33	273	30.77 %	1.17	0.06	0.12	6.36 %	0.56	36.26 %	38
☐ Walk Forward Optimiza...	0	EURUSD_fndb	H4	\$ -976.26	245	43.67 %	0.94	-0.01	-0.03	-1.32 %	0.09	1.68 %	56

We can see that Walk-Forward result ended up in loss, \$ - 976.

When we'll double click on the result in the databank, the strategy results will open.

Optimization Engine

Walk Forward Optimization results - EURUSD_fhdb, H4, 01/11/2006 - 30/06/2011, OOS: 30/06/2011 - 30/06/2011

Progress

Settings

Results

Walk-Forward result

Overview

List of trades

Equity chart

Trade analysis

Strategy settings

Source code

Overall Walk Forward Result

Fail

Test criteria	Result	Comment
1.Overall Profitability	Fail	Total profit \$ -976 < \$ 0
2.Walk-Forward Robustness	Fail	Walk-Forward Efficiency < 50 %. The system performs with rate less than 50 % of results achieved by optimization part.
3.Consistency of profits	Fail	Less than 60 % of Walk-Forward runs were profitable
4.Distribution of profits	Fail	Run # 1 contributed to more than 50 % of profits
5.Maximum drawdown	Pass	No run had drawdown bigger than 25 % of initial capital
6.Minimum trades	Fail	Run # 1 had less than 30 trades

We can see that the strategy failed in walk-forward test, based on our success criteria. On the top you have the overall result, and in the Test criteria table below you can see the results broken down by each success criteria used.

Below this table we'll see the reoptimization period that was used:

Reoptimization period is: reoptimize every 204 days on history of 475 days

We didn't specify the days exactly (we could by checking the Define specific number of days check in the Progress panel).

These values (204 days of run and 475 days of optimization were computed from the available history data because we specified we want to have 6 runs and 30% Out of sample (run) data.

Below this information we can check the results for each optimization and run period:

Optimization periods		Run periods								
Run	Period	D...	# of trades	Net profit	Drawdown	Max DD %	Winning %	Parameters		
1	19/12/2006 - 07/04/2008	475	53	\$ 939.51	404	3.93 %	35.85 %	pEMA_1=15.0, pEMA_2=20.0, Maximum Trades Per Day=0.0, Maxi...		
2	11/07/2007 - 28/10/2008	475	85	\$ 5358.5	364	2.49 %	38.82 %	pEMA_1=5.0, pEMA_2=20.0, Maximum Trades Per Day=0.0, Maxim...		
3	31/01/2008 - 20/05/2009	475	28	\$ 3039.01	864.5	6.48 %	42.86 %	pEMA_1=20.0, pEMA_2=60.0, Maximum Trades Per Day=0.0, Maxi...		
4	22/08/2008 - 10/12/2009	475	99	\$ 1690.81	2431.5	23.75 %	74.75 %	pEMA_1=15.0, pEMA_2=10.0, Maximum Trades Per Day=0.0, Maxi...		
5	14/03/2009 - 02/07/2010	475	110	\$ 3312.52	1188.89	8.1 %	77.27 %	pEMA_1=15.0, pEMA_2=10.0, Maximum Trades Per Day=0.0, Maxi...		
6	04/10/2009 - 22/01/2011	475	31	\$ 1577.5	642.4	7.21 %	29.03 %	pEMA_1=20.0, pEMA_2=60.0, Maximum Trades Per Day=0.0, Maxi...		

Optimization periods		Run periods								
...	Period	D...	# of trades	Net profit	Drawdown	Max DD %	Winning %	Parameters		
0	07/04/2008 - 28/10/2008	204	21	\$ 2999.5	220	1.99 %	47.62 %	pEMA_1=15.0, pEMA_2=20.0, Maximum Trades Per Day=0.0, Maxi...		
1	28/10/2008 - 20/05/2009	204	53	\$ -1494.09	1803.29	12.94 %	24.53 %	pEMA_1=5.0, pEMA_2=20.0, Maximum Trades Per Day=0.0, Maxim...		
2	20/05/2009 - 10/12/2009	204	25	\$ -2431.49	2454	19.72 %	12 %	pEMA_1=20.0, pEMA_2=60.0, Maximum Trades Per Day=0.0, Maxi...		
3	10/12/2009 - 02/07/2010	204	49	\$ 395.01	1188.89	10.47 %	77.55 %	pEMA_1=15.0, pEMA_2=10.0, Maximum Trades Per Day=0.0, Maxi...		
4	02/07/2010 - 22/01/2011	204	32	\$ -1789.49	1921.59	18.23 %	59.38 %	pEMA_1=15.0, pEMA_2=10.0, Maximum Trades Per Day=0.0, Maxi...		
5	22/01/2011 - 30/06/2011	160	12	\$ 404.8	323.3	3.75 %	41.67 %	pEMA_1=20.0, pEMA_2=60.0, Maximum Trades Per Day=0.0, Maxi...		

You can see that only 3 out of 6 runs ended up in profit.

We can also check the equity chart:



Interpreting the results

How should we interpret these results?

First of all, it is clear that this particular strategy will not benefit from reoptimization, at least not from the reoptimization using the periods we specified - 6 reoptimizations in 5 years, which is roughly reoptimization every 200 days.

It is possible that this strategy simply not robust enough and it will fail in the future.

But there is also a possibility that the strategy could be profitable if different, more frequent periods of reoptimizations are chosen.

How can we tell what is the best reoptimization period for strategy?

This is where [Walk-Forward Matrix](#) comes to play.

6.2.3 Walk-Forward Matrix

Walk-Forward Matrix is a powerful feature that can help you with two things:

1. Find the optimal period for strategy reoptimization

it will help you identify the best optimization frequency

2. Test the strategy for robustness

if the strategy passes Walk-Forward Matrix test it means that with the help of parameter reoptimization it is adaptable to a big range of market conditions.

Standard Walk-Forward optimization tests the strategy results if it is periodically reoptimized, let's say every 300 days.

But how do we know how often it is the best to reoptimize the strategy?

We can only guess, unless we'll use Walk-Forward Matrix that will test various combinations of reoptimization periods.

Walk-Forward Matrix example

Step 1: Configuring Walk-Forward Matrix

We'll continue with our example where we left it in the [Walk-Forward Optimization](#) part. This time we'll choose Walk-Forward Matrix option.

Optimization Settings

☐ Simple Optimization
☐ Walk-Forward Optimization
☒ **Walk-Forward Matrix**

Multi Walk-Forward Analysis

☐ Define specific days

	Start	Stop	Increment
Out of Sample %	20	30	10
Walk Forward Runs	10	40	10

In the matrix we have to specify which combinations we want to test for Walk-Forward Optimization.

All the other settings (combination of parameter values to test and success criteria) are already set by previous examples, so we can click on the **Start** button to start testing.

Computation of Walk-Forward Matrix is very time-consuming, because it has to run full Walk-Forward optimization for every combination in the matrix.

It is better to not set too many steps, at least if you want to have the results relatively quickly.

Step 2: Checking the results

When the optimization is over, we once again get a new **Walk-Forward Matrix result** in our databank.

Walk-Forward results										
☐	Load	Save	▼	Delete	Clear all	Rows: 2				
	Generation	Fit...	Symbol	TF	Net profit	# of tra...	% Wins	Profit...	Sharp...	R-Ex...
☐	Original strategy	0.48	EURUSD_fhdb	H4	\$ 2760.33	273	30.77 %	1.17	0.06	0.12
☐	Walk Forward Matrix r...	0	EURUSD_fhdb	H4	\$ 6730.71	260	52.31 %	1.59	0.13	0.28

Double-click on it to load it to the Results screen.

Optimization Engine
EURUSD_fhdb/H4 - EURUSD_fhdb, H4, 01/11/2006 - 30/06/2011, OOS: 30/06/2011 - 30/06/2011

Progress
Settings
Results

Walk-Forward result
Overview
List of trades
Equity chart
Trade analysis
Strategy settings
Source code

Multi Walk-Forward Matrix
Display: Result

Out of sample % / Runs	10	20	30	40
20	Fail	Fail	Fail	Fail
30	Fail	Fail	Fail	Fail

Walk-Forward results for **Out of sample = 30% and Runs = 40**

Test criteria	Result	Comment
1.Overall Profitability	Pass	Total profit \$ 6731 >= \$ 0
2.Walk-Forward Robustness	Fail	Walk-Forward Efficiency < 50 %. The system performs with rate less than 50 % of results achieved by optimization part.
3.Consistency of profits	Pass	More than 60 % of Walk-Forward runs were profitable
4.Distribution of profits	Pass	No individual period contributed to more than 50 % of profits
5.Maximum drawdown	Pass	No run had drawdown bigger than 25 % of initial capital
6.Minimum trades	Fail	Initial Optimization period had less than 30 trades

Reoptimization period is: reoptimize every 40 days on history of 91 days

Optimization periods
Run periods

We can see that all the combinations of WF optimization failed (according to the success criteria we defined before).

You can click on any of the cell on Multi Walk-Forward Matrix table to update the results for the selected combination. Let's choose combination 30 / 40 on the bottom right.

If you'll look at the Test criteria table, you'll see that it failed only in 2 out of 6 criteria - in Walk-Forward Robustness and number of trades.

We can switch the Display combo box to show us net profit.

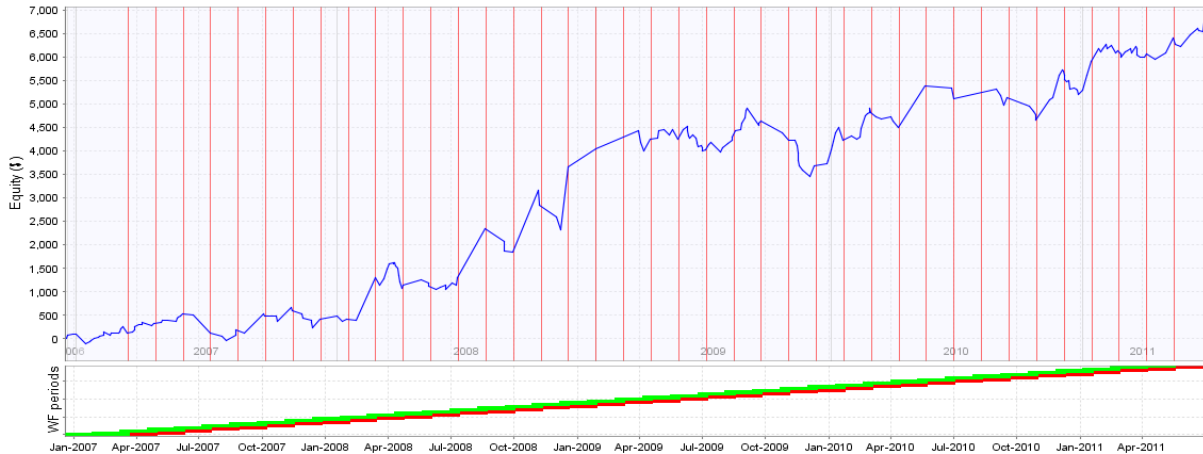
Multi Walk-Forward Matrix				
		Display: Net Profit		
Out of sample % / Runs	10	20	30	40
20	\$ 159	\$ 5031	\$ 3806	\$ 6198
30	\$ 1193	\$ 4913	\$ 4006	\$ 6731

This is not bad at all, the best result was achieved for combination of 30% Out of Sample and 40 optimization runs, which corresponds to reoptimization every 40 days with history of 91 days.

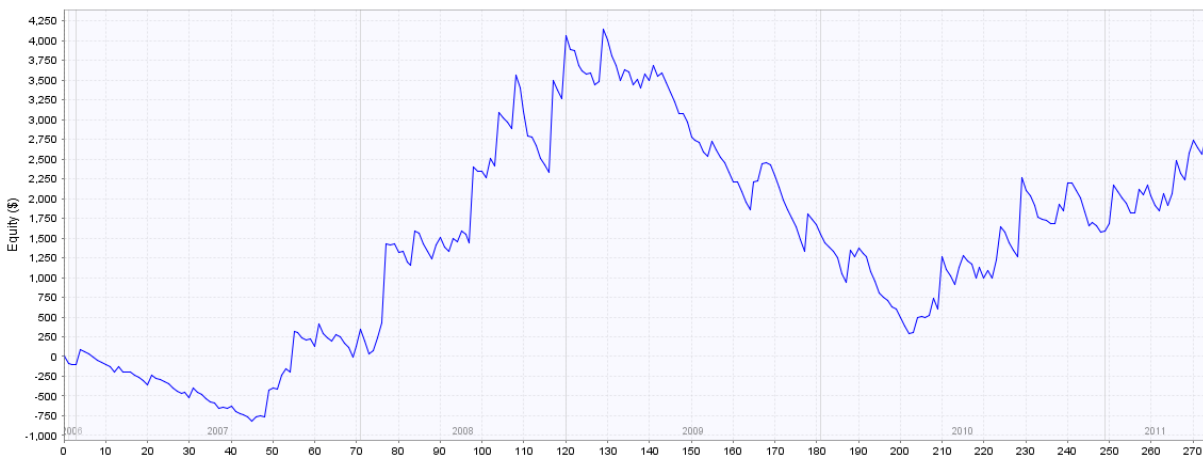
The strategy then made profit \$ 6731, while the original strategy without reoptimizations made only \$ 2760.

So it seems that frequent reoptimization does benefit the strategy.

We can look at the equity chart for this combination.



it is not bad, the chart is not ideal, but it is a big improvement from our original strategy chart that looked like this:



We can see that reoptimized strategy remained profitable for most of the time, even during the periods when the original strategy was losing.

Final conclusion

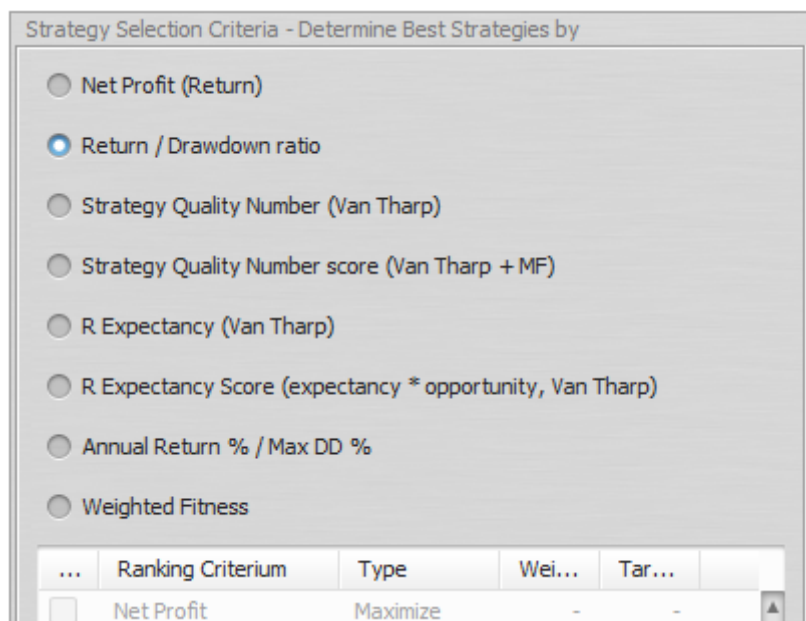
We saw that this particular strategy benefited from frequent reoptimization. It still failed our success criteria, but perhaps with some additional filter its efficiency can be improved further.

Possible improvements

There are several possible improvements of our Walk-Forward analysis, for example we can choose different steps for the parameter values.

We chose pEMA_1 to go from 5 to 20 in steps of 5, and pEMA_2 to go from 10 to 60 in steps of 10. Perhaps the Walk-Forward result would be better if both values are optimized to range from 5 to 50 with steps 5.

Another setting that influences Walk-Forward result is our choice of Selection Criteria.



In our examples we chose to select the best strategy by Return / Drawdown ratio.

Who knows, maybe the Walk-Forward optimization would have better results if we'd choose Strategy Quality Number score, or R Expectancy.

You can repeat the optimization with another Selection Criteria to check it out.

6.3 Portfolios

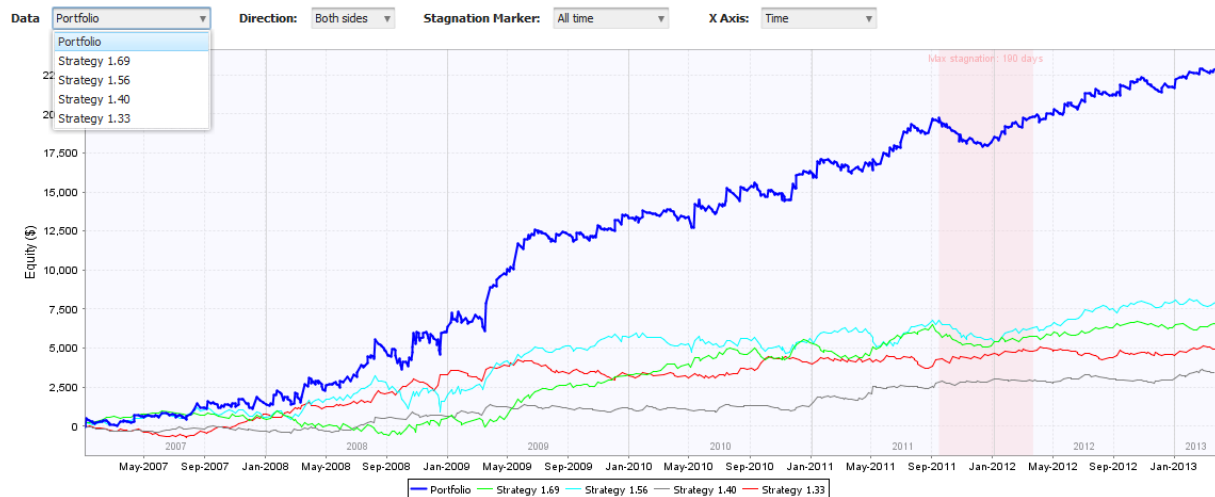
There are two possibilities how to create a portfolio:

1. Use [Additional data](#) to build/test strategy on multiple symbols or timeframes - this will create a portfolio that will contain results of the same strategy tested on different data
2. **Manually [combine strategies to a portfolio](#)** - this will create a portfolio of different strategies combined together, trading on the same or different symbols & timeframes

Portfolio results are a summary results of all the strategies it contains. For portfolios created manually (2) there is a special "Portfolio #" result created in databank that can be saved to a file for later use.

Portfolio chart

Equity chart for portfolio combines equity of the portfolio as well as equity for every strategy in the portfolio. When viewing a portfolio you can switch between portfolio chart and charts of substrategies using a combo box on the top right corner of the chart.



6.4 Strategy Editor

Note!

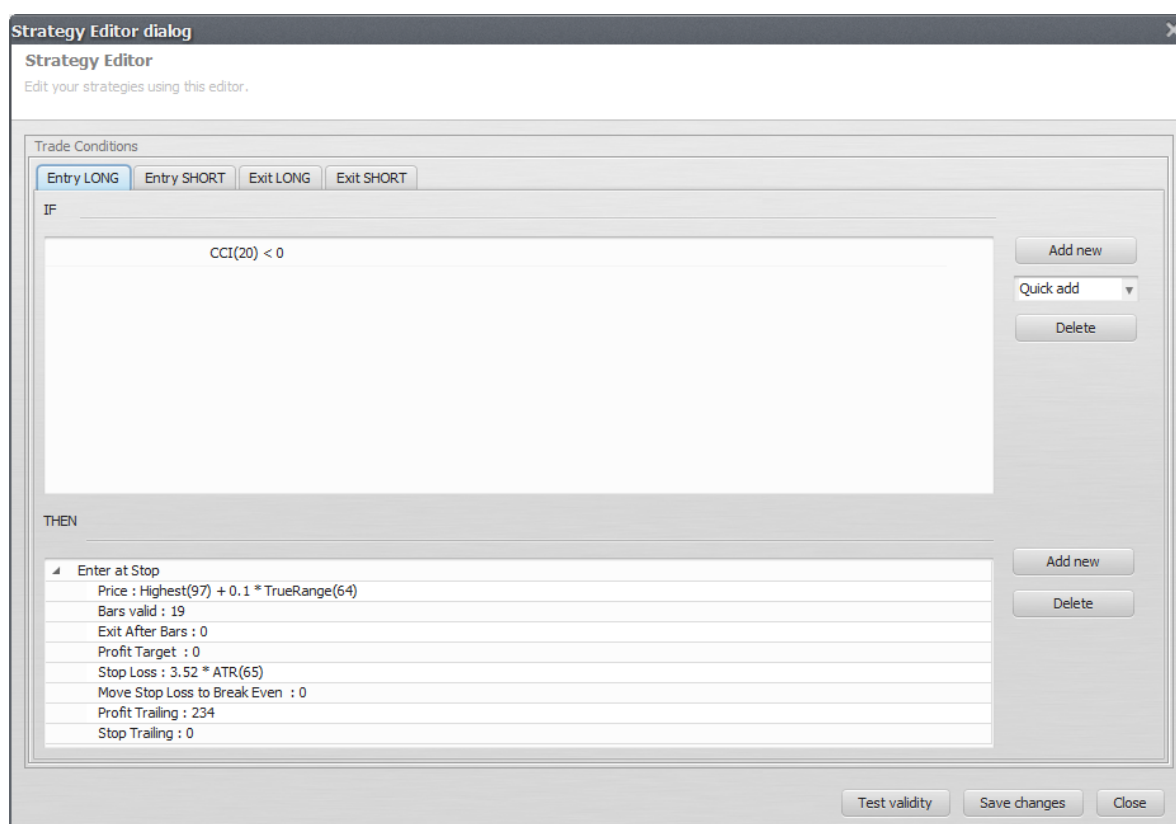
Genetic Builder strategies have a fixed format, and editor matches this format with 4 tabs, for Entry Long, Entry Short, Exit Long

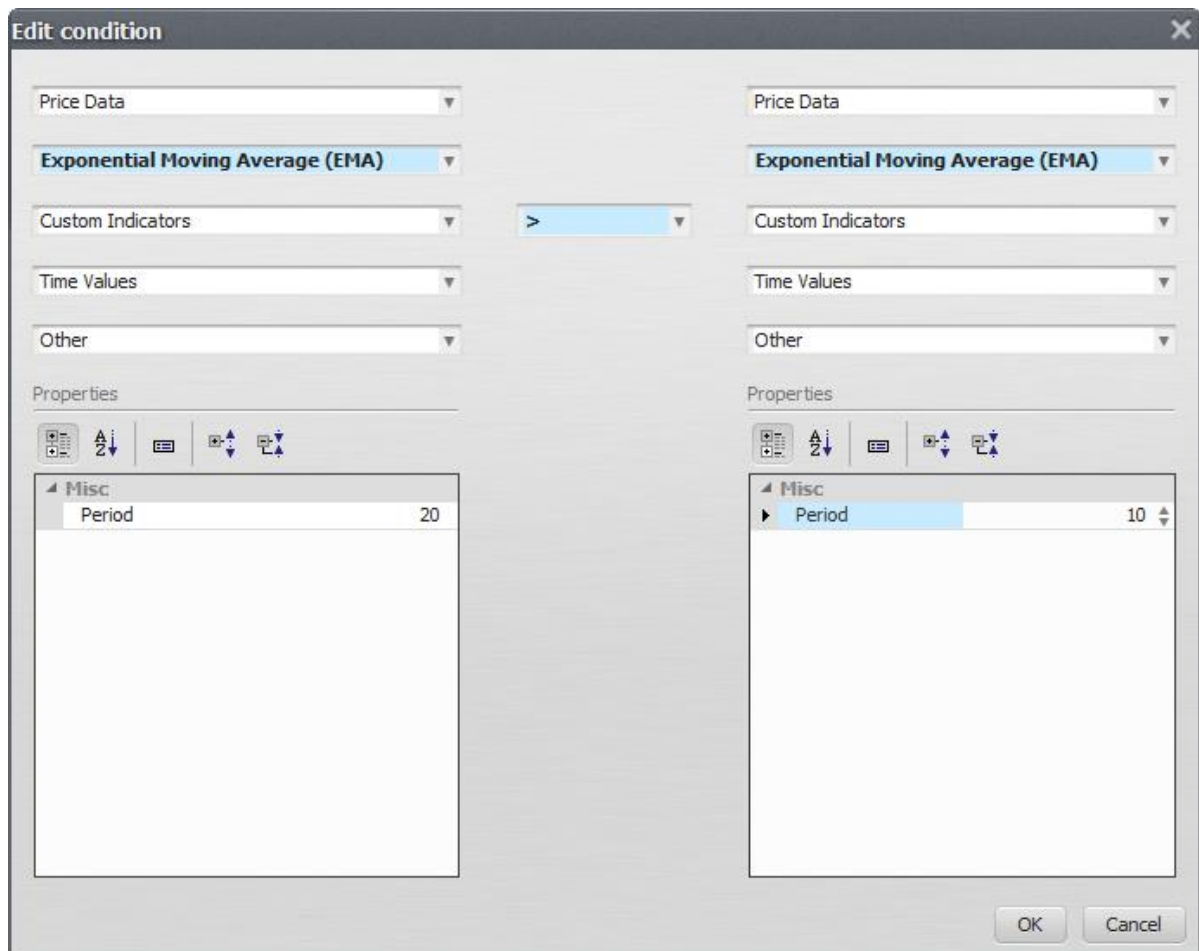
(if we want to close the trade using some rule other than Stop Loss or Profit Target) and Exit Short.

Each tab has IF and THEN part – in the IF part you define the conditions, and in the THEN part you define what should happen once the conditions are true.

To **CREATE** new strategy, go to menu File -> Create strategy or click on the icon on the toolbar.

To **EDIT** existing strategy just select it in the databank and click Edit.



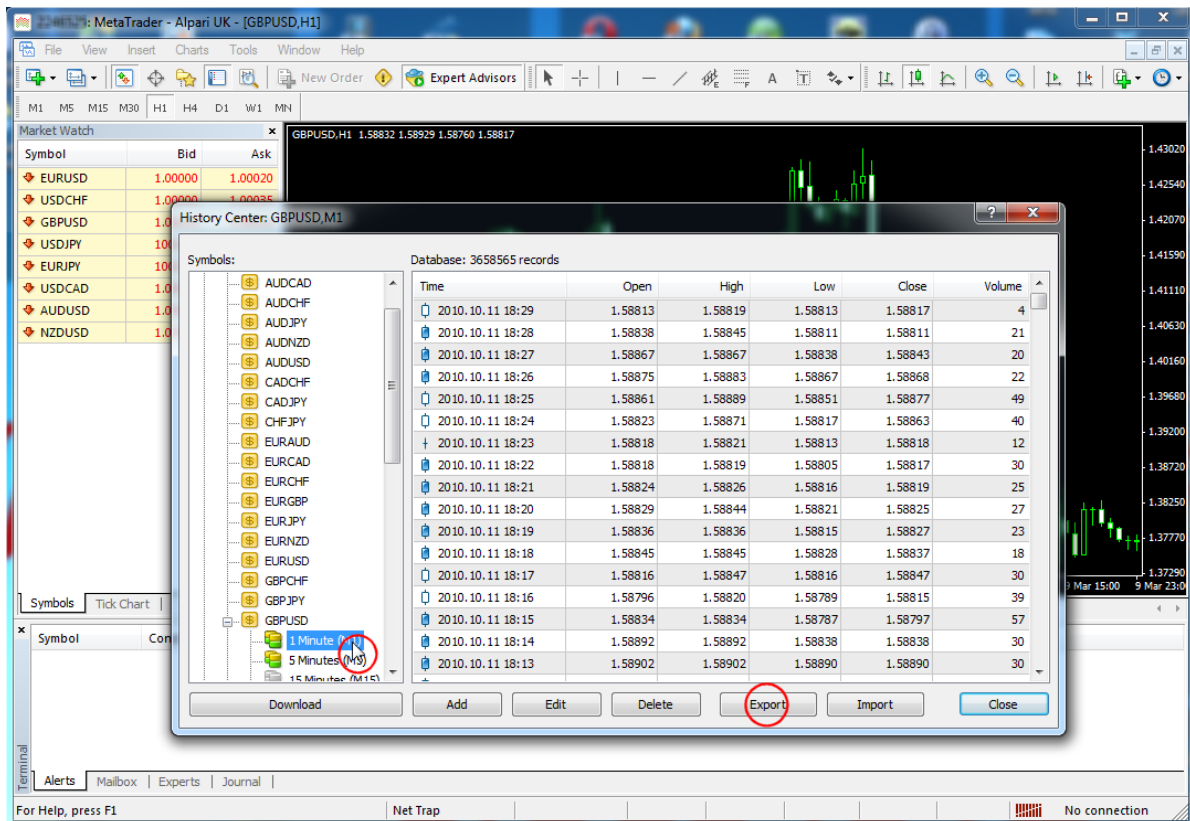


7 How to...

7.1 Import history data from MetaTrader

Step 1: Export data from Metatrader

Open your Metatrader and go to **Tools -> History Center**.



There, open the currency you want to export (for example GBPUSD) and double click on **1 Minute (M1)** so that it is refreshed on the right side of the screen.

Then just click on the **Export** button and choose your destination file.

Note

StrategyQuant supports import of only 1 Minute data, it will compute the higher timeframes automatically.

Now we have the data ready to be imported to StrategyQuant.

Step 2: Create a new symbol in StrategyQuant

We can import data to already existing symbol, but that will overwrite its data, so it is better to define a new symbol.

Start StrategyQuant, go to **Data manager** screen, **History Data** and click on **Add symbol...**

Add new symbol

Symbol Info

Symbol: GBPUSD_mt4 Symbol Full Name: GBPUSD_mt4

Pip/Tick Point: 0.0001 Pip/Tick Value in \$: 10

what is one pip unit? what is the money value of one pip?

Pip/Tick Step: 0.00001

what is minimum pip step?

Default Spread: 3 pips Cost per roundturn: 0

OK Cancel

Fill out the symbol and full name, check the default spread and tick value.

Pip/Tick point

is the value of pip. It means what number is 1 pip. is usually 0.0001. For **JPY** based pairs 1 pip is 0.01.

Pip/Tick value in \$

is a value how much is one pip worth in money, it is usually 10 for all the currencies.

Pip/Tick Step

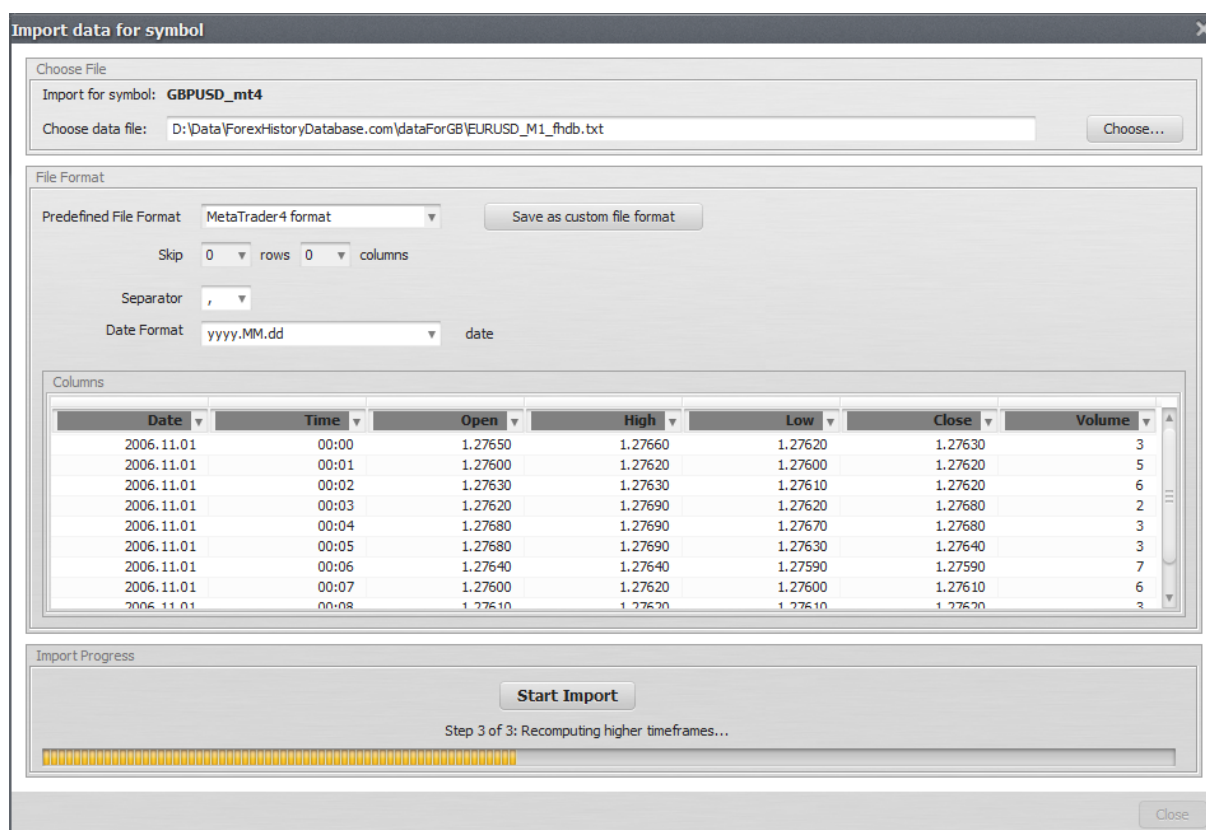
is a value by how much one pip can move. Virtually all brokers now use 5-digit data, so the value will be 0.00001 (or 0.001 for JPY based pairs).

Click OK and the new symbol will be created. The symbol doesn't have any data yet, but we are going to import them in the next step.

Step 3: Import data to StrategyQuant

Now select your new row with the symbol and click on **Import data ...** button on the top. This will open the import dialog.

Import dialog is configurable; it allows importing data from various file formats. Since we use data from MetaTrader, choose MetaTrader4 as a Predefined File Format.



Choose the data file and click on Start Import button. This will start the import process. It could take few minutes, depending on speed of your computer and data size.

When the import is finished it will display information window and asks you to close the dialog.

Now we have the new data successfully imported into StrategyQuant and we can use them for tests or new strategies generation.

7.2 Import new custom indicator

Custom indicators are indicators that run under MetaTrader4 platform and that are not implemented in StrategyQuant.

StrategyQuant is not able to run indicator code, so the only way how to work with custom indicators in SQ is by importing their data.

Note

Custom indicators are computed in MetaTrader, not in SQ. To get the correct results you have to use the same history data in both StrategyQuant and MT4.

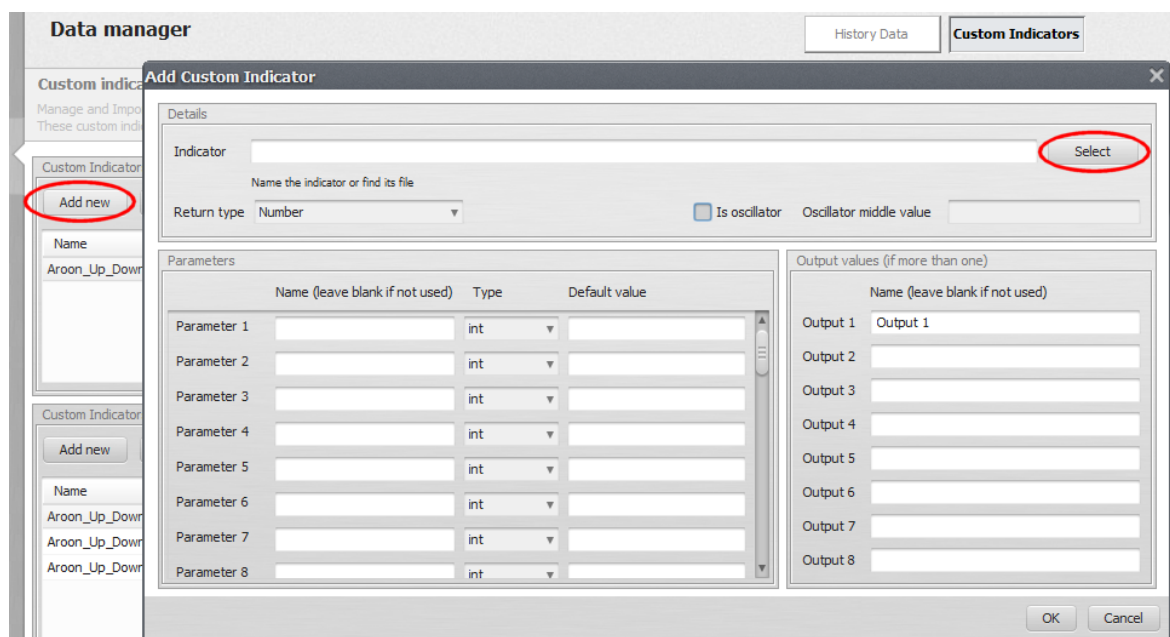
It will not work to compute your custom indicator in MetaTrader with your broker data and then use them in StrategyQuant on data from other source.

Before using custom indicators, synchronize your history data so that both SQ and MT4 use the same history.

Step 1: Define new custom indicator in StrategyQuant

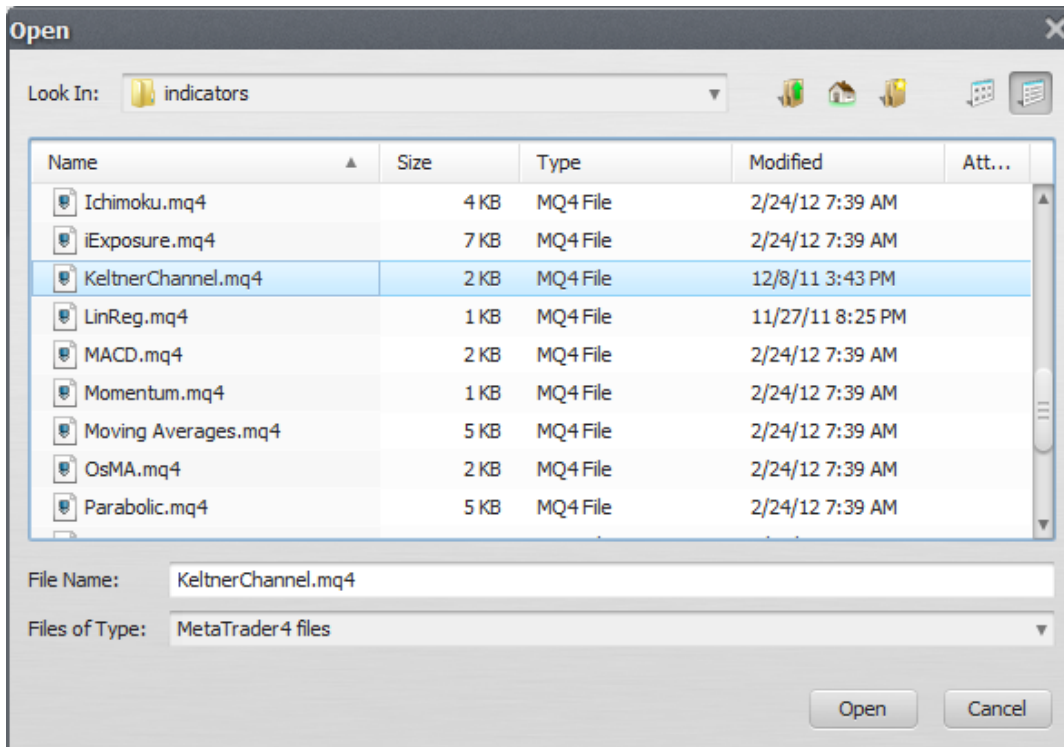
Go to Data manager -> Custom indicators and click on **Add new** in the Custom Indicators definitions table.

This will open new indicator definition dialog.

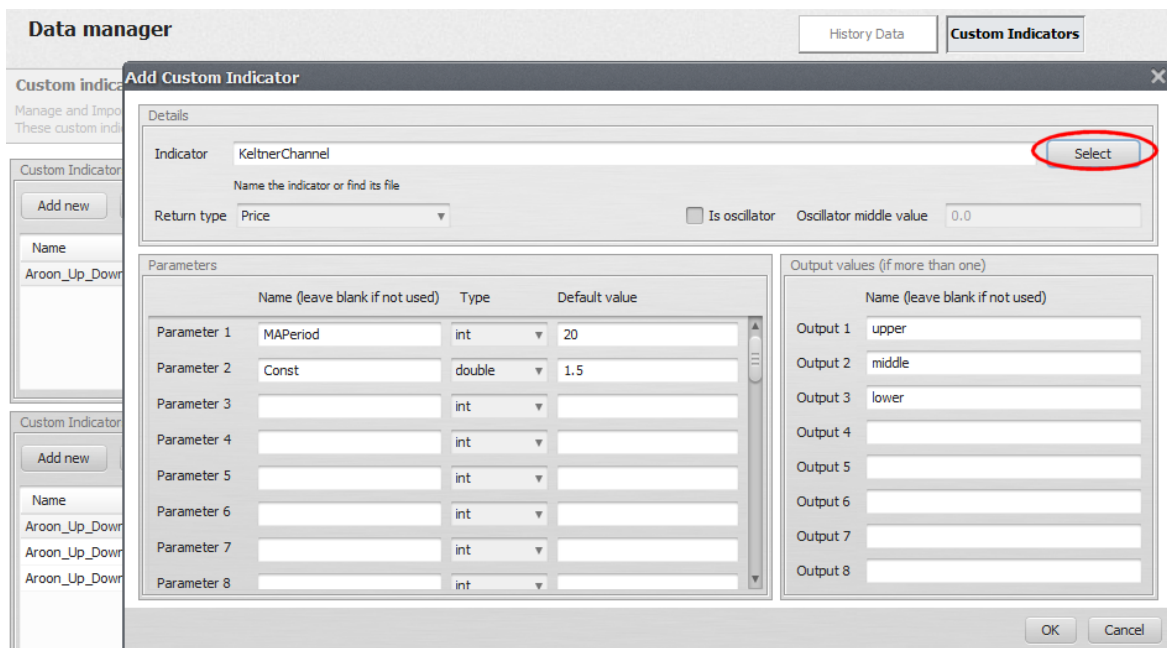


You can define your indicator manually, but if you have a source code of your indicator (.mq4 file) you can use automatic recognition.

Click on Select and find your indicator file - it must end with .mq4 extension. We'll try to import Keltner Channel indicator to make an example.



The indicator file will be loaded and parsed and all the fields in Add indicator form will be automatically filled.



We can see that StrategyQuant recognized indicator Return type as Price, it has two parameters and three output values.

Return type

is a type of return value indicator generates, it is used by StrategyQuant to properly match this custom indicator to other building blocks in the program, so that it compares price with price, and not for example price with CCI value.

It can be either:

- **Number** - if it is indicator such as CCI, RSI, MACD etc.
- **Price** - if the indicator value is price, like moving average or Bollinger Bands.
- **Price range** – if the indicator value is price range (difference between two prices), such as ATR or Bollinger Bands Range.
- **Boolean** – if the indicator returns true/false values (zero/nonzero in our case). You can have boolean indicator for example to recognize candle patterns or to implement your own simple trading rules.

How to decide the proper return type

generally, if the indicator draws its lines on the same chart as price, then its return type is price. If it draws its lines into a separate window below the main chart then it is Number, with the exception of special indicators like ATR that compute some kind of price difference or range.

Is oscillator

Some indicators like CCI or RSI are oscillators, which means that its value oscillates around some number, like CCI oscillates around 0, or RSI oscillates around 50). If the indicator is an oscillator, check this option.

Oscillator middle value

for oscillators put their middle value (the value around which they oscillate) here.

Parameters

parameters are a complete list of parameters together with their types and default values for the indicator. If you define the custom indicator manually you can check its parameters in MetaTrader.

Output values

these are the values indicator outputs. If we'll check the screenshot of Keltner Channel indicator below you can see that it outputs three values in Data window, and they correspond to upper, middle and lower line on the chart.



Click **OK** on the dialog and your new custom indicator is defined.

Step 2: Define custom indicator data

Now that we defined new custom indicator definition, we have to load the custom indicator data. The definition that we made in the first step is only a "description" of the indicator. To be able to use the indicator in your strategies, you have to configure which parameter and output values it should use and load data for this indicator computed from MetaTrader.

Click on the **Add new** button in the Custom Indicators data table.

Custom Indicators definitions

Add new Edit

Name

Aroon_Up_Down (Aroon_

KeltnerChannel (KeltnerCh

Custom Indicators data

Add new Edit

Name

Aroon_Up_Down(25,false

Aroon_Up_Down(25,false

Aroon_Up_Down(25,false

Add Custom Indicator Data

Details

Indicator KeltnerChannel Symbol EURUSD_fhdb

Output value to use upper Timeframe H1

Automatic import settings

MetaTrader directory D:\trading\Alpari NZ 1 Find

Symbol in MetaTrader EURUSD

Parameters

Name	Type	Value
MAPeriod	int	20
Const	double	1.5

OK Cancel

Here we have to choose our indicator (Keltner Channel) and choose which of its output values we want to use. We decided for upper line.

If you want to use also middle or lower line, you have to follow this step for each output you want to use.

Then you also have to choose the Symbol and Timeframe for which these indicator data will be valid. Data will be computed in MetaTrader for a given symbol and timeframe, so they will be not valid for any other symbol/timeframe. To use this custom indicator on another symbol, simply repeat this Step 2 with another symbol.

If we want to use automatic import settings (recommended) you also have to fill the path to your MetaTrader - this is MetaTrader that will be called to compute the data for this indicator. We also have to specify the exact symbol name in that MetaTrader. It is because in StrategyQuant you can use any name, like EURUSD_fhdb, but in order to run an export on MT4 we need the exact symbol name that is used in MT4.

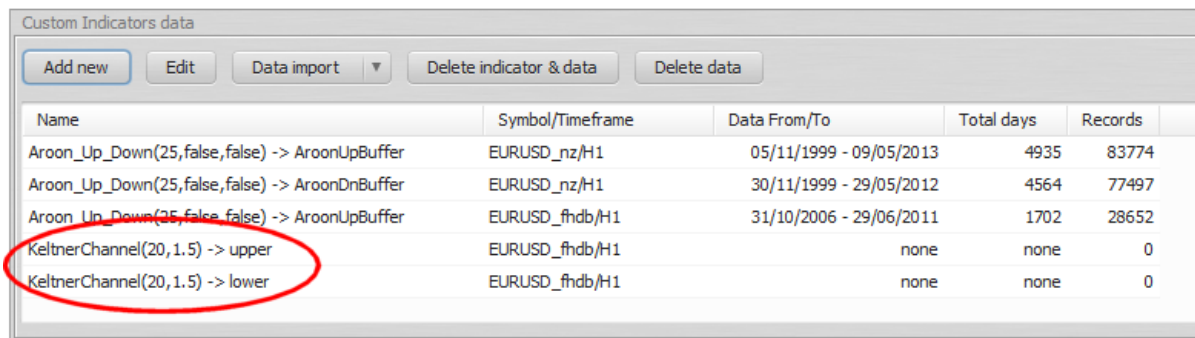
The last thing is set the indicator parameters.

Again, computed indicator data will be valid only for one combination of symbol/timeframe/output/parameters. If you want to use the custom indicator with another set of parameters, you have to define it multiple times.

When all this is done click OK to save new Custom indicator data.

We can repeat the same step for the same indicator and symbol, but this time we'll choose lower output value.

This way we'll have new custom indicator building blocks for upper and lower values of Keltner Channel indicator.

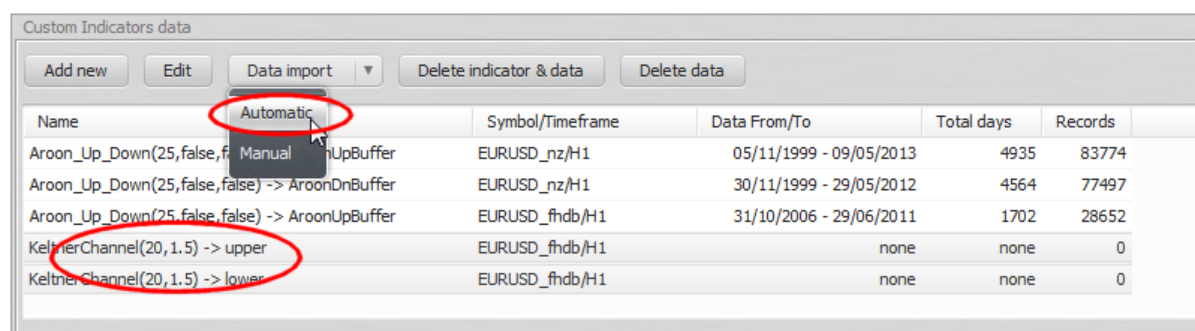


Name	Symbol/Timeframe	Data From/To	Total days	Records
Aroon_Up_Down(25,false,false) -> AroonUpBuffer	EURUSD_nz/H1	05/11/1999 - 09/05/2013	4935	83774
Aroon_Up_Down(25,false,false) -> AroonDnBuffer	EURUSD_nz/H1	30/11/1999 - 29/05/2012	4564	77497
Aroon_Up_Down(25,false,false) -> AroonUpBuffer	EURUSD_fhdb/H1	31/10/2006 - 29/06/2011	1702	28652
KeltnerChannel(20,1.5) -> upper	EURUSD_fhdb/H1	none	none	0
KeltnerChannel(20,1.5) -> lower	EURUSD_fhdb/H1	none	none	0

Step 3: Loading custom indicator data

Now we have two new custom indicators defined, but they still don't contain any data. We can use automatic import and StrategyQuant will do all the work for us.

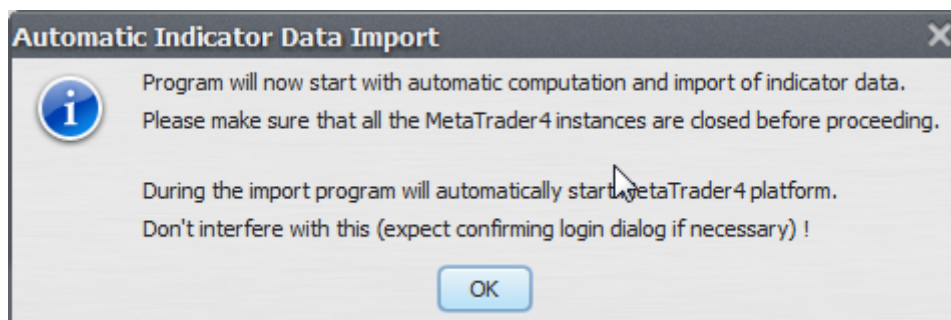
Select your two new records (hold CTRL while clicking on rows to select multiple rows at the same time), and then choose **Data import -> Automatic**



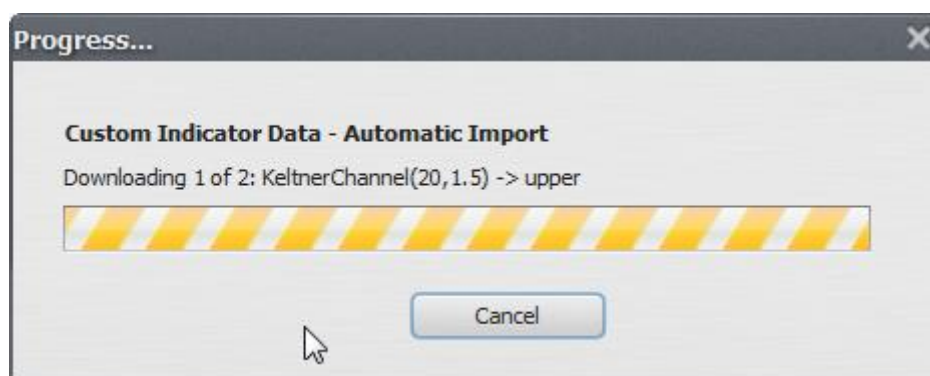
Name	Symbol/Timeframe	Data From/To	Total days	Records
Aroon_Up_Down(25,false,false) -> AroonUpBuffer	EURUSD_nz/H1	05/11/1999 - 09/05/2013	4935	83774
Aroon_Up_Down(25,false,false) -> AroonDnBuffer	EURUSD_nz/H1	30/11/1999 - 29/05/2012	4564	77497
Aroon_Up_Down(25,false,false) -> AroonUpBuffer	EURUSD_fhdb/H1	31/10/2006 - 29/06/2011	1702	28652
KeltnerChannel(20,1.5) -> upper	EURUSD_fhdb/H1	none	none	0
KeltnerChannel(20,1.5) -> lower	EURUSD_fhdb/H1	none	none	0

Note

Before running automatic import please close your MetaTrader, otherwise the import will fail.



Program will now automatically start MetaTrader and initiate custom indicator computation, data export and its import to StrategyQuant.



All this is automatic, you don't need to interfere with it in any way.

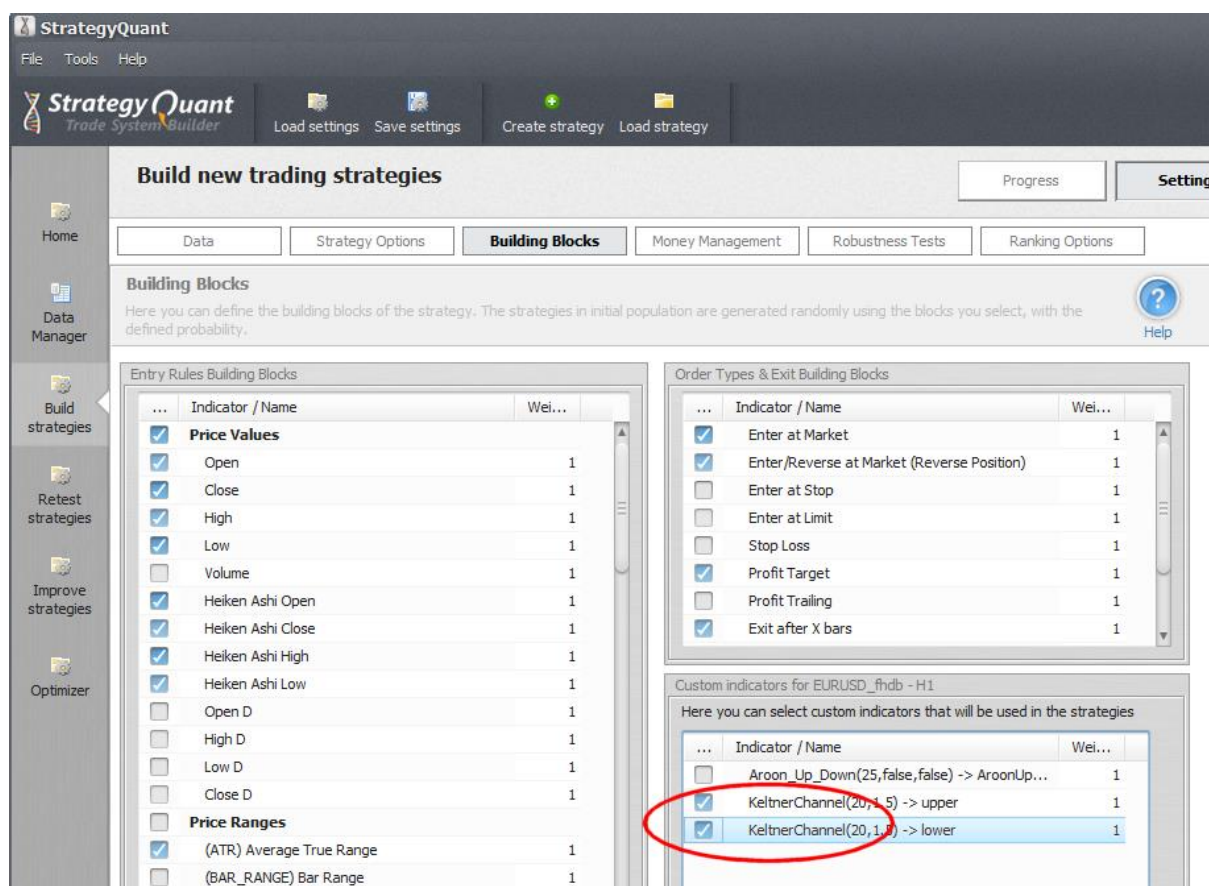
When the process finishes, you'll see that the indicators now contain imported values:

Custom Indicators data				
Add new Edit Data import Delete indicator & data Delete data				
Name	Symbol/Timeframe	Data From/To	Total days	Records
Aroon_Up_Down(25,false,false) -> AroonUpBuffer	EURUSD_nz/H1	05/11/1999 - 09/05/2013	4935	83774
Aroon_Up_Down(25,false,false) -> AroonDnBuffer	EURUSD_nz/H1	30/11/1999 - 29/05/2012	4564	77497
Aroon_Up_Down(25,false,false) -> AroonUpBuffer	EURUSD_fhdb/H1	31/10/2006 - 29/06/2011	1702	28652
KeltnerChannel(20,1.5) -> upper	EURUSD_fhdb/H1	31/10/2006 - 29/06/2011	1702	28652
KeltnerChannel(20,1.5) -> lower	EURUSD_fhdb/H1	31/10/2006 - 29/06/2011	1702	28652

Step 4: Using your new custom indicators

Steps 1 - 3 have to be done only once for every new custom indicator and symbol/timeframe/output/parameters you want to add to StrategyQuant.

When the indicator data are successfully imported you can use them in your building blocks.



To use your new custom indicators in your strategies, you have to check them in Building Blocks -> Custom indicators table.

Note

The custom indicators are valid only for the symbol/timeframe they were computed, in our example only for EURUSD_fhdb / H1.

If you want to use them for another symbol or timeframe, you have to repeats steps 1 - 3.

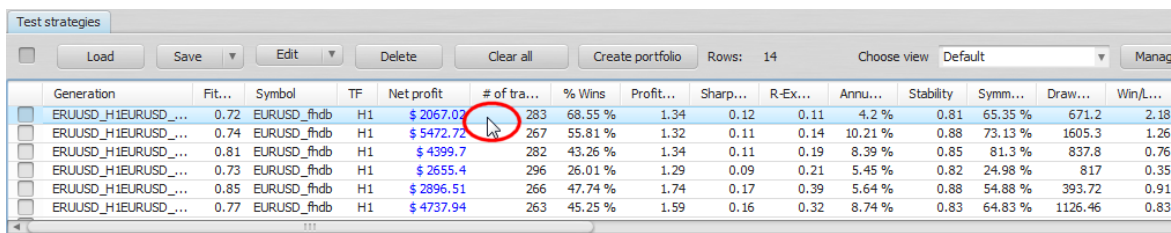
7.3 Export strategy from StrategyQuant and test or trade it in MetaTrader

When you generate some strategies and find the ones you would potentially like to use in real trading, it is time to test them in MetaTrader.

StrategyQuant normally saves strategies in its own proprietary .str file format, which is not readable by MetaTrader.

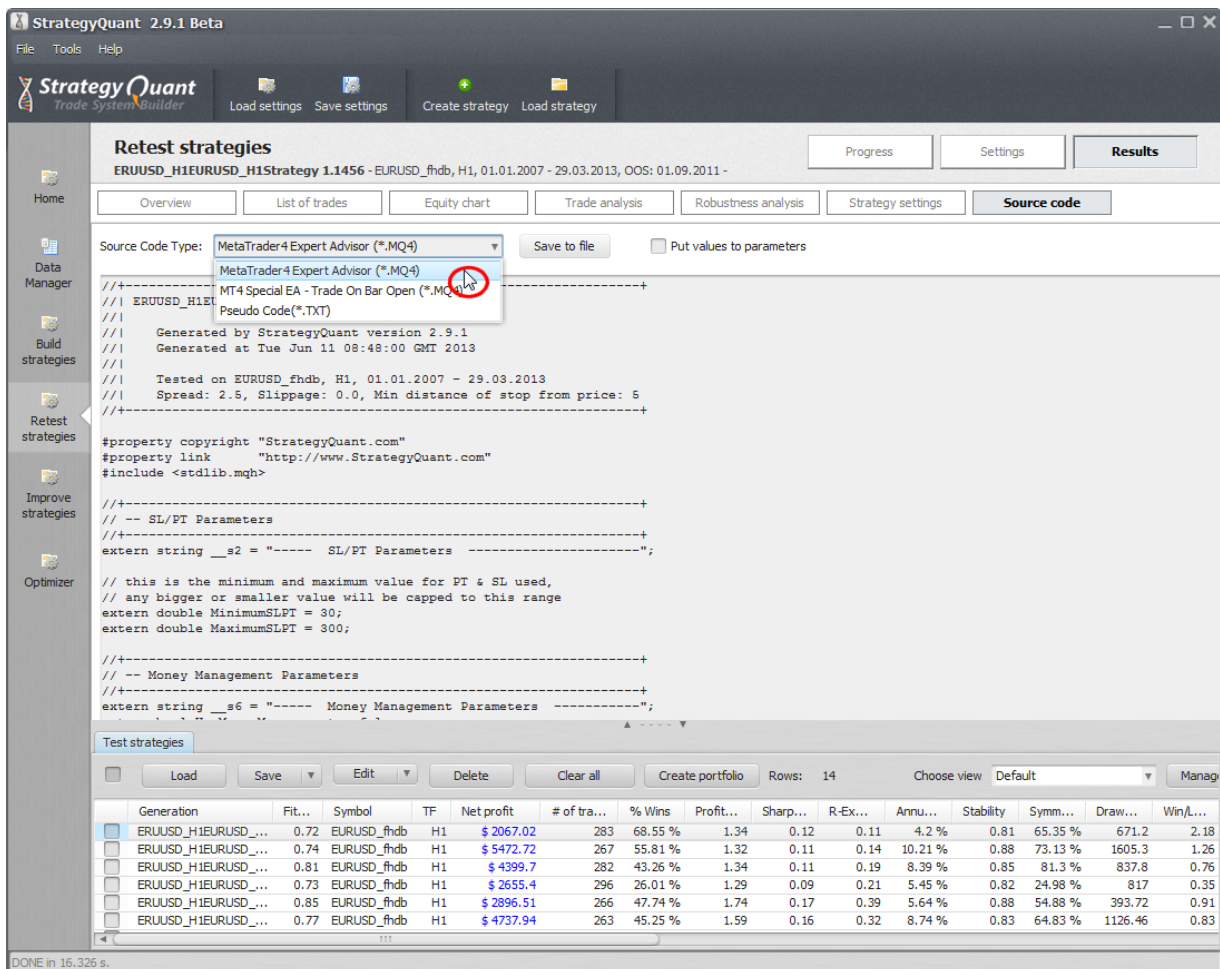
In order to test strategies in MT4 you have to export its source code in MQL format

This is simple, go to the databank and find the strategy you want to use. Double-click on it, which opens it in the Result details window above the Databank.



Generation	Fit...	Symbol	TF	Net profit	# of tra...	% Wins	Profit...	Sharp...	R-Ex...	Annu...	Stability	Symm...	Draw...	Win/L...
ERUSD_H1EURUSD_...	0.72	EURUSD_fhdb	H1	\$ 2067.02	283	68.55 %	1.34	0.12	0.11	4.2 %	0.81	65.35 %	671.2	2.18
ERUSD_H1EURUSD_...	0.74	EURUSD_fhdb	H1	\$ 5472.72	267	55.81 %	1.32	0.11	0.14	10.21 %	0.88	73.13 %	1605.3	1.26
ERUSD_H1EURUSD_...	0.81	EURUSD_fhdb	H1	\$ 4399.7	282	43.26 %	1.34	0.11	0.19	8.39 %	0.85	81.3 %	837.8	0.76
ERUSD_H1EURUSD_...	0.73	EURUSD_fhdb	H1	\$ 2655.4	296	26.01 %	1.29	0.09	0.21	5.45 %	0.82	24.98 %	817	0.35
ERUSD_H1EURUSD_...	0.85	EURUSD_fhdb	H1	\$ 2896.51	266	47.74 %	1.74	0.39	0.39	5.64 %	0.88	54.88 %	393.72	0.91
ERUSD_H1EURUSD_...	0.77	EURUSD_fhdb	H1	\$ 4737.94	263	45.25 %	1.59	0.16	0.32	8.74 %	0.83	64.83 %	1126.46	0.83

There, go to Source code tab and switch the source code to MetaTrader4 Expert Advisor. This will load the MT4 code of the strategy.



StrategyQuant 2.9.1 Beta

File Tools Help

StrategyQuant Trade System Builder

Load settings Save settings Create strategy Load strategy

Retest strategies

ERUSD_H1EURUSD_H1Strategy 1.1456 - EURUSD_fhdb, H1, 01.01.2007 - 29.03.2013, OOS: 01.09.2011 -

Progress Settings Results

Overview List of trades Equity chart Trade analysis Robustness analysis Strategy settings Source code

Source Code Type: MetaTrader4 Expert Advisor (*.MQ4) Save to file Put values to parameters

```

//+-----+
//| ERUSD_H1EURUSD_H1Strategy 1.1456 - EURUSD_fhdb, H1, 01.01.2007 - 29.03.2013, OOS: 01.09.2011 - |
//+-----+
//|
//| Generated by StrategyQuant version 2.9.1
//| Generated at Tue Jun 11 08:48:00 GMT 2013
//|
//| Tested on EURUSD_fhdb, H1, 01.01.2007 - 29.03.2013
//| Spread: 2.5, Slippage: 0.0, Min distance of stop from price: 5
//+-----+
//|
//| #property copyright "StrategyQuant.com"
//| #property link      "http://www.StrategyQuant.com"
//| #include <stdlib.mqh>
//+-----+
//|
//| -- SL/PT Parameters
//+-----+
extern string __s2 = "----- SL/PT Parameters -----";
//+-----+
//|
//| this is the minimum and maximum value for PT & SL used,
//| any bigger or smaller value will be capped to this range
extern double MinimumSLPT = 30;
extern double MaximumSLPT = 300;
//+-----+
//|
//| -- Money Management Parameters
//+-----+
extern string __s6 = "----- Money Management Parameters -----";
//+-----+

```

Test strategies

Load Save Edit Delete Clear all Create portfolio Rows: 14 Choose view Default Manage

Generation	Fit...	Symbol	TF	Net profit	# of tra...	% Wins	Profit...	Sharp...	R-Ex...	Annu...	Stability	Symm...	Draw...	Win/L...
ERUSD_H1EURUSD_...	0.72	EURUSD_fhdb	H1	\$ 2067.02	283	68.55 %	1.34	0.12	0.11	4.2 %	0.81	65.35 %	671.2	2.18
ERUSD_H1EURUSD_...	0.74	EURUSD_fhdb	H1	\$ 5472.72	267	55.81 %	1.32	0.11	0.14	10.21 %	0.88	73.13 %	1605.3	1.26
ERUSD_H1EURUSD_...	0.81	EURUSD_fhdb	H1	\$ 4399.7	282	43.26 %	1.34	0.11	0.19	8.39 %	0.85	81.3 %	837.8	0.76
ERUSD_H1EURUSD_...	0.73	EURUSD_fhdb	H1	\$ 2655.4	296	26.01 %	1.29	0.09	0.21	5.45 %	0.82	24.98 %	817	0.35
ERUSD_H1EURUSD_...	0.85	EURUSD_fhdb	H1	\$ 2896.51	266	47.74 %	1.74	0.39	0.39	5.64 %	0.88	54.88 %	393.72	0.91
ERUSD_H1EURUSD_...	0.77	EURUSD_fhdb	H1	\$ 4737.94	263	45.25 %	1.59	0.16	0.32	8.74 %	0.83	64.83 %	1126.46	0.83

DONE in 16.326 s.

Click on Save to file button, and in the file dialog find the folder where your MetaTrader is installed (for example C:\Program Files\Alpari MT).

In this folder you have to go inside the **experts** directory and save the strategy source code there.

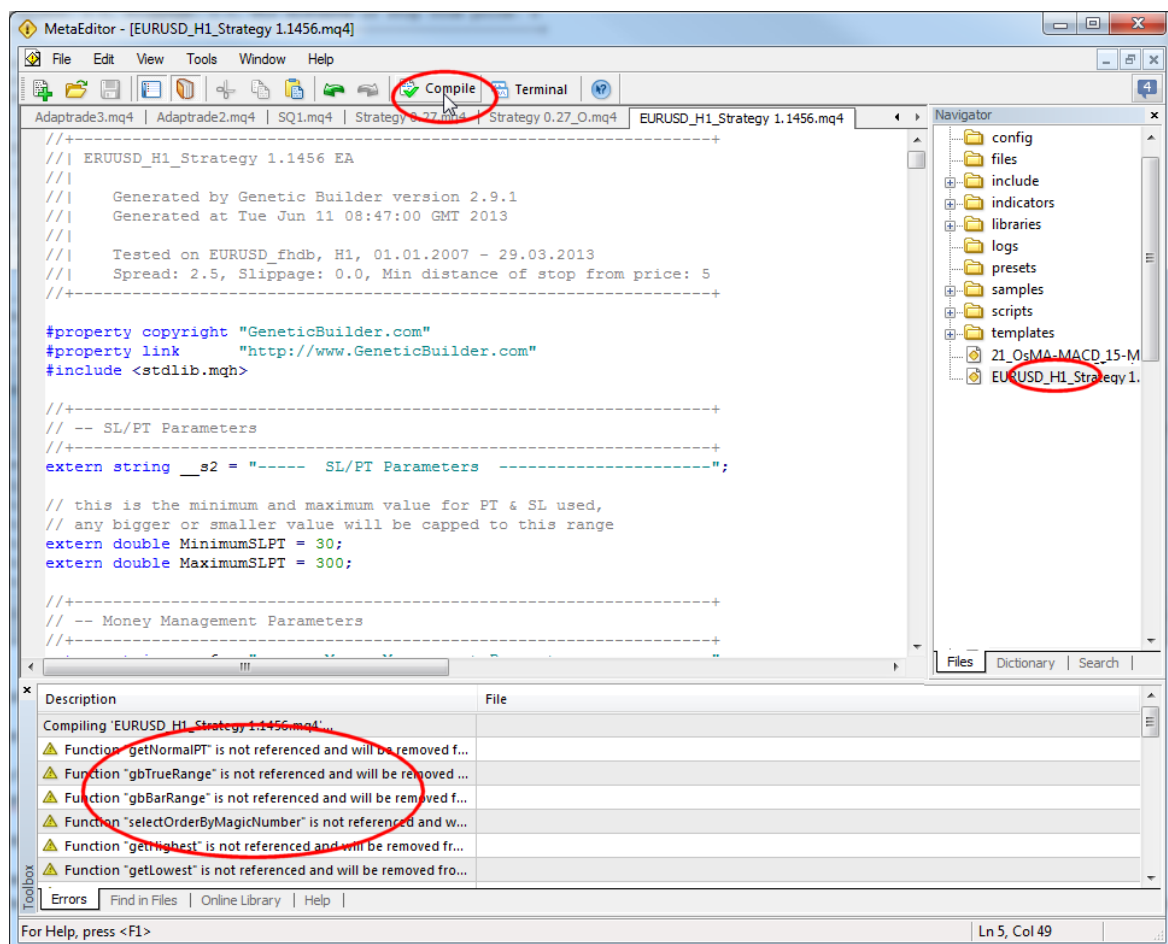
So the full path of the file will be for example

C:\Program Files\Alpari MT\experts\ERUUSD_H1_Strategy 1.1456.mq4

Now the strategy is copied to Metatrader. You can open MetaTrader now.

In Metatrader go to menu **Tools -> MetaQuotes Language Editor**, or press **F4**. This will open the language editor.

On the right side of the editor you'll have a list of strategies that are in the **experts** folder. Double-click on our strategy to open it in the editor window and then click on **Compile** on the top toolbar.



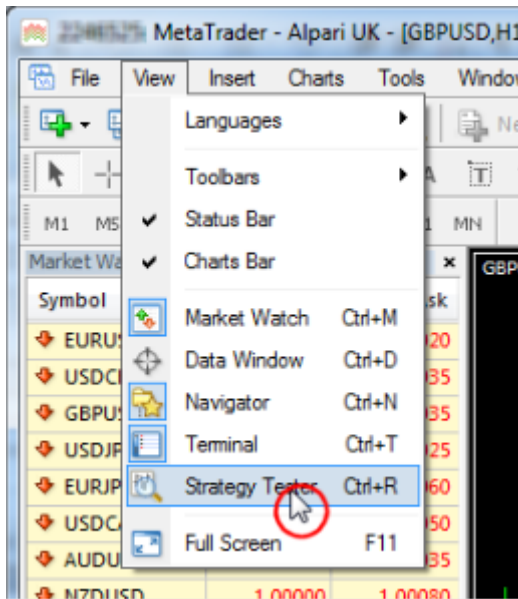
The strategy will be compiled and now it is ready for backtest or running live.

Note - Compilation warnings are normal

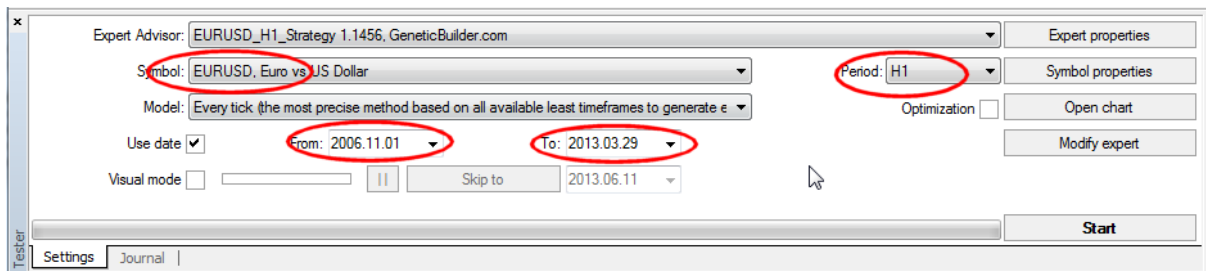
Please note that there are some compilation warnings on the bottom. These warnings are normal and they don't influence the strategy work.

There are simply some functions that are not used in the strategy and MetaTrader is informing you about that.

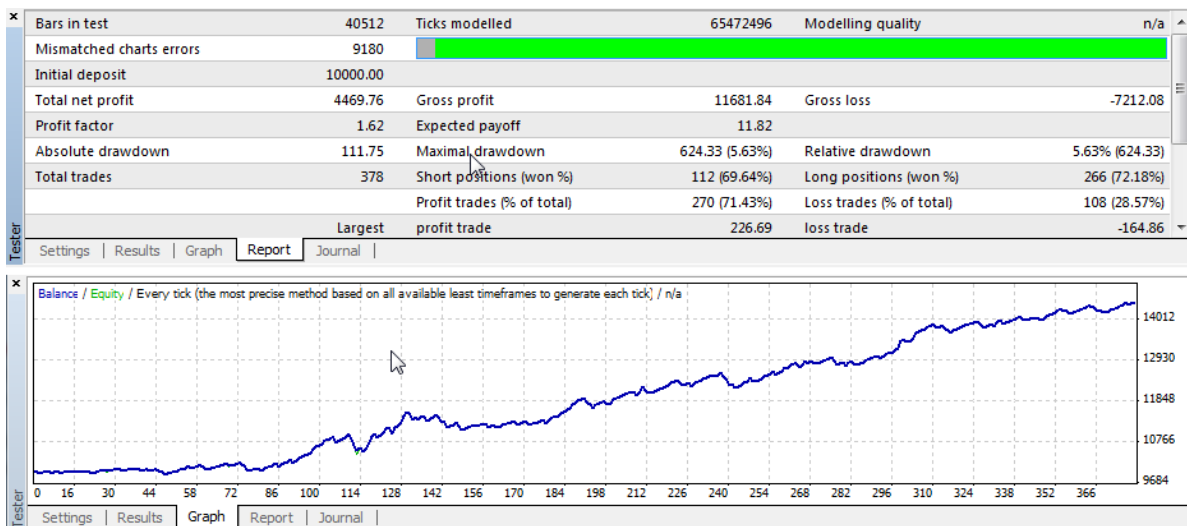
Now that the strategy is compiled, it is ready to be backtested. You can close the **MetaEditor**, go to the main MetaTrader screen and open **Strategy Tester**.



This will open the **Strategy Tester** dialog on the bottom and you can run the backtest.



Make sure you select the correct Expert Advisor, Symbol, Timeframe and Date From and To and then click on the **Start** button. The test will start and after a while you'll get the results:



Explanation of small differences in backtests

If you'll compare test results in StrategyQuant and in MetaTrader, you'll see that on some cases the backtesting results are not the same.

The results can differ slightly or significantly - depending on the type of strategy.

Backtesting algorithm used in StrategyQuant is very accurate, but it is not exactly the same algorithm used in MetaTrader, so it produces slightly different result.

The important thing here is to understand that both testing algorithms are only approximations, one isn't superior to the other.

7.4 Translate program to another language

The program is by default in English, but it supports translation to another languages. All you have to do is to create a new language file for your language in **{StrategyQuant}/lang** folder.

There is an English.lng file that contains all language strings used in the program.

All you have to do is to create a copy of this file, rename it for example to Deutsch.lng (for german) and then open it in plain text editor like Notepad and add translation for every string.

The file format is simple - if is language string in English then equals sign and after that should be language file in another language.

For example:

version=Your translated string

Homepage=Your translated string

Get Support=Your translated string

etc.

When you created new language file just restart the program and you should see your new language in menu **View -> Language**

Notes for translation!

- If you use any special characters the file has to be saved in UTF-8 format
- For strings that contain special marks like %d, %s these marks will be replaced by the appropriate value by the program.
You should use them in the correct places also in your translated text
- The length of language string should be not much longer than the length of original string. Use abbreviation if needed.
The reason is that there is only a limited place on the user interface for the texts. If you'll use too long text it might not fit to the label.
- When you create language file for a new language share it with others! Send it to us and we will include it to the next update of the program.

8 Appendix

8.1 Ranking criteria

Net Profit

Net Profit is a total profit/loss the strategy produced.

Drawdown, Max DD %

Drawdown is the measure of the decline from a historical peak in running cumulative profit of the strategy. Max DD % is the maximum percentage drawdown of the strategy.

Stagnation, % Stagnation

stagnation is a maximum number of days during which strategy stagnates - which means it doesn't make a new high on the equity. Obviously you'd want strategy with as little stagnation as possible.

% of Wins

percentage of winning trades

Annual % Return

average annual percentage return of the strategy. If it is 30 %, it means the strategy makes 30% every year in average.

Annual % Return / Max DD %

important ratio between annual percentage and maximum percentage drawdown. You'd want this number to be as high as possible.

Avg Daily, Monthly, Yearly profit

average profit for given period in \$

Return/DD Ratio

a very good measure of strategy quality, the higher the number, the bigger are profits in relation to maximum drawdown.

Expectancy

performance metrics developed by Van Tharp, it is computed as (percentage wins * average win) - (percentage losses * average loss)

R-Expectancy

performance metrics developed by Van Tharp, it gives you the average profit value related to average risk (R) that you can expect from a system over many trades.

You can find more at: <http://www.vantharp.com/tharp-concepts/expectancy.asp>

R-Expectancy Score

standard R-Expectancy doesn't consider the length of testing period and number of trades produced. There is a difference is you'll make let's say \$2000 per year by making 10 trades, or by making 100 trades.

R-Expectancy Score adds a score for trades frequency. It is computed as: $\text{R-Expectancy} * \text{averageTradesPerYear}$

SQN (System Quality Number)

performance metrics developed by Van Tharp, it is the measure of the quality of a trading system.

You can find more at: <http://www.vantharp.com/tharp-concepts/sqn.asp>

Standard interpretation of SQN is:

- Score: 1.6 – 1.9 Below average, but trade-able
- Score: 2.0 – 2.4 Average
- Score: 2.5 – 2.9 Good
- Score: 3.0 – 5.0 Excellent
- Score: 5.1 – 6.9 Superb
- Score: 7.0 - Keep this up, and you may have the Holy Grail.

SQN Score (System Quality Number Score)

just like in case of R-Expectancy, SQN doesn't consider the length of testing period and number of trades produced. In fact it is more favorable for systems that produce more trades, without considering the length of the testing period.

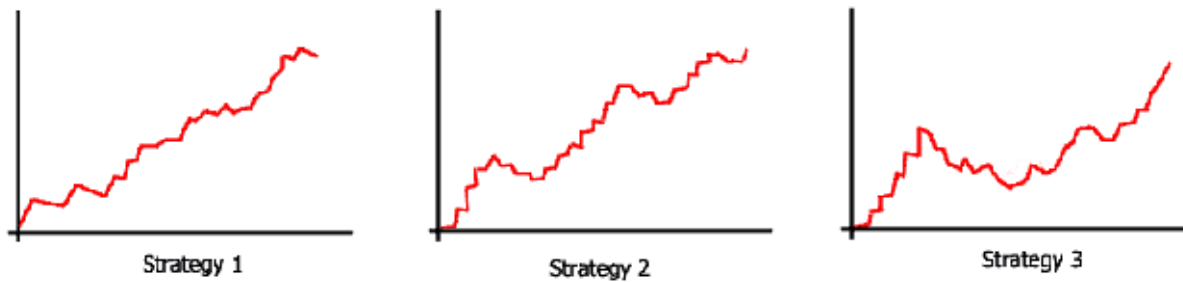
It is computed as: $\text{SQN} * (\text{averageTradesPerYear} / 100)$

Number of trades

simply a number of trades of this strategy in the backtest. You can use this criterion to approximate some preset value you would like to achieve (for example total 100 trades). The value of this criterion will be bigger for strategies closer to the desired number of trades.

Stability

a special value in range from 0 (worst) to 1 (best) that measures how stable is growth of the equity chart.



You can see three sample strategies on the picture above. All of them finished with the same profit, but Strategy 1 was growing almost linearly (very stable), Strategy 2 was growing with occasional big drawdowns (less stable) and Strategy 3 was moving up and down (very little stability of growth).

Stability is a value that is quite good in representing a "quality" of the trading strategy and it can be used as the only one or the main criterion to compute the total strategy *Fitness*.

Symmetry

criterion to maximize strategy symmetry. Symmetry value is in %, and it is measuring how much is the Profit/Loss for Long direction similar to Short direction.

For example, if strategy makes \$600 on Long trades, and \$400 on Short trades, symmetry in this case is 66%. (\$400 is 66% from \$600).

If the strategy makes the same profit on both directions, the symmetry will be 100%. If one of the directions produces loss or 0 profit, the symmetry will be 0%.

Win/Loss Ratio

criterion to maximize the ratio of winning trades vs. losing ones

Average Win, Average Loss

criterion to maximize average win or minimize average loss per trade

Average Bars in Trade

criterion to minimize the average number of bars the trade is open

Average Bars Win, Average Bars Loss

criterion to minimize the average number of bars for the winning/losing trade

Degrees of Freedom

in a trading system relates to the number of criteria that are used to filter price action and/or volume and determine entry points.

The more criteria and variables used to determine entry timing points, the fewer degrees of freedom the system will have and vice versa.

Degrees of freedom are computed from strategy complexity and number of trades. The simpler the strategy is, the more degrees of freedom it will have. For this property, the bigger value is better.

Complexity

measures the complexity of the strategy. It is simply a count of all the indicators, prices, operators and other building blocks that are used in the strategy. The higher this number is, the more “complicated” the strategy is.

For this property, the smaller value is better.

Both properties can be used also in the ranking to influence the Fitness function.

8.2 Entry and exit types

Profit Target (PT)

is the opposite of the Stop Loss, it can be used to take the profit once the price moves in your favor. As in the example above, you can set up a profit target to close the position once it makes for example 100 pips.

Exit After X Bars

this is very simple exit condition. If used, the strategy will close the trade after given number of bars (time periods on a given timeframe).

This is many times a simple, but effective way to close the trade when the market moves sideways or when we want to lock in the profit.

Move Stop Loss to Break-Even

another simple but quite powerful condition. It tells the strategy to move protective Stop Loss to the entry price (break even) when the given profit is reached. This will not close the trade, so the strategy will be still in position, but there is zero risk after this move. Because SL is at the entry price we'll not lose anything even if the trade reverses and goes against us.

This condition is evaluated only on open bar (not on every tick).

Profit Trailing

a simple trailing stop that trails the Stop Loss specified distance from highest achieved profit. This condition is evaluated on open bar (not on every tick).

Stop Trailing

more advanced trailing stop that can use a value of indicator or price (Open, High, Low, Close) to trail the Stop Loss. For example the rule can trail SL at $\text{Lowest}(20) + 20$ pips.

This condition is evaluated on open bar (not on every tick).

Exit Rule (Price + Operators + Indicators, ...)

it is a special type of exit where strategy has not only rules for entering the trade (Entry rules) but also rules for exiting it (Exit rules). Exit rules look as same as entry rules - it is a combination of all the selected building blocks(indicators, operators, price values).

Example of entry and exit rule pair:

Entry Rule - if $\text{CCI} > 0$ Enter at Market

Exit Rule - if RSI crosses above 100 and we are in long position then Close Trade

Note on Stop Loss / Profit Target/Move Stop Loss to BE

Both Stop Loss and Profit Target can have either fixed value in pips (like 30, 50 or 100 pips), or they can be based on ATR (Average True Range), for example $2 \times \text{ATR}$.

This means that the actual value of the SL/PT will be 2 times the value of average true range at the time the trade was opened.

ATR based SL/PT have the advantage that they "adapt" to changed market conditions and changes of volatility. Some strategies (but not all of them) can be more profitable when they use ATR based SL/PT instead of fixed one.

To enable/disable ATR for SL/PT, you can enable/disable the Average True Range indicator in the Entry Building Blocks.

9 Final Words

Congratulations on completing the guide to StrategyQuant. I hope you'll like the program and you'll use it as an important part of your trading.

StrategyQuant is very new program, and it is underactive development. This means you can look forward for new exciting features that will move the program functionality even further.

I'm open to a new ideas or suggestions, so if you are missing something in StrategyQuant or if you think something can work in a different way, don't hesitate to let me know.

I wish you many successful trades,

Mark Fric

P.S. I would love to hear your success stories, so please let me know at <mailto:support@strategyquant.com>